The University of Saskatchewan
Department of Computer Science

Technical Report #2004-01

University of Saskatchewan Dept. of Computer Science
Theory & Formal Bioinformatics Group
Technical Report 2004-a

Families of Languages Defined by Ciliate Bio-operations. [1]

by

Mark Daley[a], Lila Kari[b], Ian McQuillan[b]

[a]Department of Computer Science,
University of Saskatchewan,
Saskatoon, Saskatchewan, S7N 5A9, Canada
daley@cs.usask.ca

[b]Department of Computer Science,
University of Western Ontario,
London, Ontario, N6A 5B7, Canada
{lila,imcquill}@csd.uwo.ca

[1]This research was funded in part by Natural Sciences and Engineering Research Council of Canada grants (Helmut Jürgensen), (Lila Kari), the Canada Research Chair Program (Lila Kari) and institutional grants of the University of Saskatchewan (Mark Daley).

A version of this paper is to appear in *Theoretical Computer Science*.

ABSTRACT

We investigate families of languages defined by closure under operations generalized from models of gene descrambling in stichotrichous ciliates. We specifically consider languages that are closed under the synchronized insertion and deletion operations as well as languages closed under the hi (hairpin inversion) operation. Biologically, this studies sets of genes that cannot be further descrambled. In addition, we show that every trio closed under hairpin inversion is also closed under the dlad (double loop with alternating direct pointers)-excision/reinsertion bio-operation.

1 Introduction

The stichotrichous ciliates are a family of single-celled organisms with a unique genetic computational mechanism. The germline genome of these ciliates consists of genes stored in a scrambled form which the cell must descramble in order to create a functional gene capable of generating proteins.

Each ciliate cell contains both a functional macronucleus, with unscrambled versions of the ciliate's genes and an inert micronucleus, containing scrambled versions of the genes. When two ciliates conjugate, their respective macronuclei are destroyed, and they exchange the genetic material in their micronuclei. The cell is then faced with the daunting task of unscrambling the micronuclear genes in order to re-construct a functional macronucleus. For details on the biological aspects of this operation, the reader is referred to [19, 20].

It was first noted by Kari and Landweber in [18] that it is natural to view this process as a computation. A collection of papers describing the abstract properties of such a computation followed, primarily based upon two principal models of gene scrambling. The first model, proposed by Kari and Landweber is based on one inter-molecular, and one intra-molecular operation and has been investigated in [17, 5, 3]. The second model, proposed by Ehrenfeucht, Prescott and Rozenberg is based upon three intra-molecular operations and has been studied in [8, 7, 9, 4].

Previous work, particularly on the Kari-Landweber model, has considered the gene descrambling problem from the point of view of operations on well-known language families. Specifically, the closure properties and solvability of language equations involving standard language families was investigated.

In this paper we take a different approach. Rather than considering properties of known families of languages under ciliate bio-operations, we will define new families of languages based upon closure under ciliate bio-operations. We will then investigate the formal language theoretic properties of these new language families in an attempt to better understand the structure and properties of ciliate genomes. This is a natural investigation biologically, as it corresponds with the study of sets of genes which cannot be further descrambled using iterated application of the operation. In this sense, no further evolution is possible using the chosen operation.

2 Preliminaries

We begin with a brief review of the notation and concepts used throughout the paper. Let $\mathbb{N}$ be the set of positive integers and let $\mathbb{N}_0$ be the set of nonnegative integers. Let X be a set and let $k \in \mathbb{N}$. We denote by $[X]^k$ the set of all k-tuples $(x_1, \ldots, x_k)$ where $x_i \in X$.

We refer the reader to [15] for language and automata theory preliminaries. Let $\Sigma = \{a_1, \ldots, a_k\}$ be a finite alphabet where $k \in \mathbb{N}$. We denote by Σ^* and Σ^+ the sets of words and non-empty words, respectively, over Σ. We denote the empty word by λ. Let $w \in \Sigma^*$. Then $|w|$ denotes the length of w, and for each $a \in \Sigma, |w|_a$ denotes the number of occurrences of a in w. Let $\Psi_{(a_1,\ldots,a_k)}$, or Ψ when $(a_1, \ldots, a_k)$ is understood be the mapping from Σ^* into $[\mathbb{N}_0]^k$ defined by $\Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_k})$ for each $w \in \Sigma^*$. We call this function a Parikh mapping and we call $\Psi(w)$ *the Parikh vector* of w. Let alph(w) be the set of letters from Σ occurring in w which we extend to languages L in the natural way. Let w^R be the word obtained from w by reversing the positions of the letters. A language is any subset of Σ^*. Let $L \subseteq \Sigma^*$. We denote the complement of L, $\Sigma^* - L$ by $\overline{L}$.

We will next define an a-transducer. Intuitively, it is a nondeterministic gsm that allows output on a λ input. They are also referred to as rational transducers. An a-transducer is a 6-tuple $M = (Q, \Sigma_1, \Sigma_2, \delta, q_0, F)$, where Q is the finite state set, Σ_1 is the input alphabet, Σ_2 is the output alphabet, δ is a finite subset of $Q \times \Sigma_1^* \times \Sigma_2^* \times Q$, $q_0 \in Q$ is the initial state and $F \subseteq Q$ is the set of final states. We say that M is λ-free if $\delta \subseteq Q \times \Sigma_1^* \times \Sigma_2^+ \times Q$. Let $\vdash$ be the relation on $Q \times \Sigma_1^* \times \Sigma_2^*$ defined by letting $(q, xw, z_1) \vdash (p, w, z_2)$ for each $w \in \Sigma_1^*$ if $(q, x, y, p) \in \delta$ and $z_2 = z_1 y$. A triple (q, w, z) represents the fact that M is in state q, with w the input still to be read, and z the accumulated output. Let $\vdash^*$ be the reflexive, transitive closure of $\vdash$. Let $M = (Q, \Sigma_1, \Sigma_2, \delta, q_0, F)$ be an a-transducer. For each word $w \in \Sigma_1^*$, let $M(w) = \{z \mid (q_0, w, \lambda) \vdash^* (q, \lambda, z) \text{ for some } q \in F\}$. For every set $L \subseteq \Sigma_1^*$, let $M(L) = \bigcup_{w \in L} M(w)$. The mapping M is called an a-transducer mapping or a-transduction.

A *trio* is a language family (where a language family is defined as in [10]) closed under λ-free homomorphism, inverse homomorphism and intersection with regular sets. It is known that every trio is closed under λ-free a-transductions. A *full trio*[1] is a trio closed under arbitrary homomorphism. It is known that every full trio is closed under arbitrary a-transductions and hence arbitrary gsm mappings. We refer the reader to [10, 2] for the study of AFL's.

Let $k \in \mathbb{N}$. A *one-way, nondeterministic k-pushdown machine* is a six tuple $M = (Q, \Sigma, q_0, F, \Gamma, \delta, Z_0)$ where Q is the finite state set, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the final state set, Σ is the input alphabet, Γ is the finite pushdown alphabet, $Z_0 \in \Gamma$ is the end-marker and δ (the transition function) is a mapping from $Q \times (\Sigma \cup \{\lambda\}) \times [\Gamma]^k$ into finite subsets of $Q \times [\Gamma^*]^k$. Further, we will assume that if $(q, z_1, \ldots, z_k) \in \delta(q_1, a, y_1, \ldots, y_k)$ then $|z_i| \leq 2$ for all i and $Z_0 \in \text{alph}(z_i)$ if and only if $z_i = Z_0 z, Z_0 \notin \text{alph}(z)$ and $y_i = Z_0$.

If $|\delta(q, a, z_1, \ldots, z_k) \cup \delta(q, \lambda, z_1, \ldots, z_k)| \leq 1$ for each $q \in Q, a \in \Sigma$ and $z_1, \ldots, z_k \in$

[1] A full trio is also referred to as a cone.

Γ, then M is said to be deterministic.

An instantaneous description of $M = (Q, \Sigma, q_0, F, \Gamma, \delta, Z_0)$ is a $k+2$ tuple, $(q, w, \gamma_1, \ldots, \gamma_k)$, where $q \in Q$ is the current state, $w \in \Sigma^*$ is the remaining input and $\gamma_i \in \Gamma^*$ is the current contents of the i^{th} pushdown. Let $\vdash$ be the derivation relation defined on $Q \times \Sigma^* \times [\Gamma^*]^k$ such that $(q, aw, \alpha_1 y_1, \ldots, \alpha_k y_k) \vdash (p, w, \alpha_1 z_1, \ldots, \alpha_k z_k)$ if $(p, z_1, \ldots, z_k) \in \delta(q, a, y_1, \ldots, y_k)$ for $a \in \Sigma \cup \{\lambda\}$. Let $\vdash^*$ be the reflexive, transitive closure of $\vdash$. We define the language accepted by M, $L(M) = \{w \mid (q_0, w, Z_0, \ldots, Z_0) \vdash^* (q_f, \lambda, \gamma_1, \ldots, \gamma_k), q_f \in F, \gamma_i \in \Gamma^* \text{ for each } i\}$.

If $|\Gamma| = 2$, (the end-marker Z_0 plus one other pushdown letter), we say that M is a one-way nondeterministic k-counter automaton (or a multicounter automaton). We say that the i^{th} pushdown is j-reversal-bounded if the i^{th} pushdown makes at most j alternations between nondecreasing and nonincreasing its size on any computation of any input. If every pushdown of M is j-reversal-bounded, then we say that M is j-reversal-bounded or just reversal-bounded. The notion of reversal-bounded pushdown automata, originally referred to as finite turn pushdown automata, was introduced in [11] and also studied in [1].

For example, one-way, deterministic 2-counter languages have the same power as a Turing Machine [15]. Also, one-way nondeterministic 1-pushdown automata accept exactly the context-free languages and one-way nondeterministic 1-reversal-bounded, 1-pushdown automata accept exactly the linear languages.

3 Synchronized Insertion and Deletion Closed Languages

We now define the ciliate bio-operations of synchronized insertion, synchronized deletion and synchronized bi-deletion from [5].

The synchronized insertion operation inserts a word of the form vx into a word of the form uxw, using x as a pointer sequence, to create the word $uxvxw$. The second operation, synchronized deletion, deletes a word of the form vx from a word of the form $uxvxw$ to create the new word uxw.

Definition 3.1 *Let Σ be an alphabet and $\alpha, \beta \in \Sigma^*$.*

1. *The synchronized insertion of β into α is defined as: $\alpha \oplus \beta = \{uxvxw \mid \alpha = uxw, \beta = vx, x \in \Sigma^+, u, v, w \in \Sigma^*\}$.*

2. *The synchronized deletion of β from α is defined as: $\alpha \ominus \beta = \{uxw \mid \alpha = uxvxw, \beta = vx, x \in \Sigma^+, u, v, w \in \Sigma^*\}$.*

3. *The synchronized bi-deletion of β from α is defined as:* $\alpha \boxminus \beta = \{w \mid \alpha = xayaz, \beta = xaz, w = ya, a \in \Sigma,\ x, y, z \in \Sigma^*\}$.

We extend each of the operations above to binary operations on languages in the natural way.

We recall the following lemma from [5], that shows that it is only necessary to consider "pointer" sequences of length one.

Lemma 3.1 *Let Σ be an alphabet. For any $\alpha, \beta \in \Sigma^*$,*

1. $\alpha \oplus \beta = \{u'av'aw' \mid \alpha = u'aw', \beta = v'a, a \in \Sigma, u', v', w' \in \Sigma^*\}$.
2. $\alpha \ominus \beta = \{u'aw \mid \alpha = u'av'aw, \beta = v'a, a \in \Sigma, u', v', w \in \Sigma^*\}$.

We begin by defining the synchronized insertion closure of a language.

Definition 3.2 *Let Σ be an alphabet. We define the synchronized insertion closure over Σ (we omit Σ if the alphabet is understood) of a language $L \subseteq \Sigma^*$ as $sins(L) = \{\beta \in \Sigma^* \mid \forall \alpha \in L,\ \alpha \oplus \beta \subseteq L\}$. Moreover, we say a language L is sins-closed over Σ (we omit Σ if the alphabet is understood) if and only if $L \subseteq sins(L)$.*

We now give some examples of sins-closed languages and some languages which are not sins-closed.

Example 3.1

1. $sins(\Sigma^*) = \Sigma^*$.
2. $sins(L_{ab}) = L_{ab}$ *where* $L_{ab} = \{x \in \Sigma^* \mid |x|_a = |x|_b \geq 0\}$.
3. $L = \{a^n b^n \mid n \geq 0\}$, $sins(L) = \{\lambda\}$.
4. *If* $L_1 = \{a^2\}^*$ *then* $sins(L_1) = L_1$. *In addition, if* $L_2 = aL_1$ *then* $sins(L_2) = L_1$.
5. $L = b^*ab^*$, $sins(L) = b^*$.
6. *Let Σ be an alphabet and let $a, b \in \Sigma$. If $L = a\Sigma^*b$ then $sins(L) = \Sigma^*$.*

To better understand the properties of sins-closure, we introduce the notion of density.

Definition 3.3 *A language L is dense if for all $w \in \Sigma^*$, there exist $x, y \in \Sigma^*$ such that $xwy \in L$.*

Proposition 3.1 *Let Σ be a non-empty alphabet and* $\mathrm{alph}(L) = \Sigma$. *If $sins(L) = \Sigma^*$, then L is dense. Furthermore, the converse is not true. (L dense $\not\Rightarrow sins(L) = \Sigma^*$).*

Proof. For proving density, we must consider both nonempty and empty values of w.

If $w = w'a \in \Sigma^+ \subseteq sins(L), a \in \Sigma$, then for all $u = xaz \in L$, $xawz = xaw'az$ is in L. Since $a \in \mathrm{alph}(L)$, there must exist such a $u \in L$. Thus, there exists $xa, z \in \Sigma^+$ such that $xawz \in L$. If $w = \lambda$ then clearly there exists $x, y \in \Sigma^*$ such that $xy \in L$ since $\mathrm{alph}(L) = \Sigma \neq \emptyset$ and thus $L \neq \emptyset$.

Therefore L is dense.

To see that the converse is not true, consider the language $L_{ab} = \{x \in \Sigma^* \mid |x|_a = |x|_b \geq 0\}$, $\Sigma = \{a, b\}$. Clearly, L_{ab} is dense since for any word $w \in \Sigma^*$, we can find $x, y \in \Sigma^*$ to ensure the number of a's and b's in w is balanced. However, $sins(L_{ab}) = L_{ab}$ and thus the implication does not hold as $\Sigma^* \neq L_{ab}$. ■

We now give a characterization of sins-closure in terms of synchronized bi-deletion.

Proposition 3.2 *Let Σ be an alphabet and $L \subseteq \Sigma^*$. Then*

$$sins(L) = \overline{(\overline{L} \boxminus L)}.$$

Proof. "$\subseteq$" Suppose $x \in \overline{sins(L)}$, then for all $u \in L$, $u \oplus x \subseteq L$. Assume, by way of contradiction, that $x \notin \overline{(\overline{L} \boxminus L)}$. It must then be the case that $x \in \overline{L} \boxminus L$ and $x \in x'ayax'' \boxminus x'ax''$, $x'ayax'' \in \overline{L}$, $x'ax'' \in L$, $x = ya$. There exists a word $x'ax'' \in L$ such that $x'ax'' \oplus x \not\subseteq L$ and thus $x \notin sins(L)$, a contradiction.

"$\supseteq$" Suppose $x \in \overline{(\overline{L} \boxminus L)}$. Assume $x \notin sins(L)$. Then there must exist $u \in L$ such that $u \oplus x \not\subseteq L$. Let $u = u'au''$ and $x = x'a$, then $u'ax'au'' \notin L$ and thus $x = x'a \in \overline{L} \boxminus L$ (as $u'ax'au'' \in \overline{L}$ and $u'au'' \in L$), a contradiction. The proposition now follows. ■

Next, we define the synchronized deletion closure of a language and demonstrate a similar characterization of the closure in terms of synchronized bi-deletion.

Definition 3.4 *Let Σ be an alphabet. We define the synchronized deletion closure over Σ (we omit Σ if the alphabet is understood) of a language $L \subseteq \Sigma^*$ as $sdel(L) = \{x \in \Sigma^* \mid \forall u \in L, u \ominus x \subseteq L\}$. Furthermore, we say a language L is sdel-closed over Σ (we omit Σ if the alphabet is understood) if and only if $L \subseteq sdel(L)$.*

Proposition 3.3 *Let Σ be an alphabet and $L \subseteq \Sigma^*$. Then*

$$sdel(L) = \overline{(L \boxminus \overline{L})}.$$

Proof. "$\subseteq$" Suppose $x \in sdel(L)$ which implies for all $u \in L$, $u \ominus x \subseteq L$. Assume $x \notin \overline{(L \boxminus \overline{L})}$, equivalently, $x \in (L \boxminus \overline{L})$ and specifically $x \in \alpha a x' a \beta \boxminus \alpha a \beta$, $\alpha a x' a \beta \in L$, $\alpha a \beta \in \overline{L}$. Then $x = x'a$ and thus $\alpha a \beta \in L \ominus x \subseteq L$, a contradiction since $\alpha a \beta \in \overline{L}$.

"$\supseteq$" Suppose $x \in \overline{(L \boxminus \overline{L})}$. Assume $x \notin sdel(L)$. There must then exist $u \in L$ such that $u \ominus x \not\subseteq L$. Then u can be written as $u = u'ax'au'' \in L$, $x = x'a$ where $u'au'' \in u'ax'au'' \ominus x$ and $u'au'' \notin L$. But then $x \in L \boxminus \overline{L}$, a contradiction, and the proposition follows. ■

In the case when we start with a sins-closed language, we give a simple characterization of whether the language is also sdel-closed.

Proposition 3.4 *Let $L \subseteq \Sigma^+$ be an sins-closed language. Then L is sdel-closed if and only if $L \ominus L = L$.*

Proof. "$\Rightarrow$"

"$\supseteq$" Follows trivially from the definition of sdel-closure.

"$\subseteq$" Let $u = u'a \in L$. As L is sins-closed, we have $uu \in u'a \oplus u'a \subseteq L$ and thus $u \in L \ominus L$.

Consequently $L = L \ominus L$.

"$\Leftarrow$" Trivial, as clearly $L = L \ominus L \Rightarrow L$ is sdel-closed. ■

Let us denote the family consisting of all languages L whereby there exists a finite alphabet Σ, $L \subseteq \Sigma^*$ such that L is sins-closed over Σ by **SINS**. We define the family of all sdel-closed languages, **SDEL** similarly. Notice that **SINS** is equal to the family of languages such that $\emptyset, \{\lambda\}$ are in the family and if Σ is a finite alphabet with $\text{alph}(L) = \Sigma$ and L is sins-closed, then L is in the family. Similarly for **SDEL**.

We will now give the closure properties of the family **SINS**. It is not closed under most of the classical operations.

Proposition 3.5 *The family of synchronized-insertion-closed languages is not closed under (i) union, (ii) catenation, (iii) homomorphism, (iv) intersection with a regular set, (v) Kleene closure (vi) complement or (vii) inverse homomorphism.*

Proof. (i) Let $\Sigma = \{a, b\}$. Consider the languages $(ab)^*, a^* \in$ **SINS**. The language $(ab)^* \cup a^*$ is not in **SINS**.

(ii) $a^*, b^* \in$ **SINS** but $a^* \cdot b^* = a^*b^* \notin$ **SINS**.

(iii) The language d^* is clearly in **SINS**. Consider the homomorphism from $\{d\}^*$ into $\{a, b, c\}^*$ defined by $h : h(d) = cacbc$. Then $h(d^*) = (cacbc)^* \notin$ **SINS**.

(iv) Consider $L_{ab} = \{x \in \Sigma^* \mid |x|_a = |x|_b \geq 0\}$ over Σ. $L_{ab} \in$ **SINS** and a^*b^* is a regular language but $L_{ab} \cap a^*b^* = \{a^n b^n \mid n \geq 0\}$ which is not in **SINS**.

(v) The language $L = (aaa)^* \cup (bbb)^*$ over Σ is in **SINS** and $aaabbb \in L^*$. However, $aaabaaabbbbb \in aaabbb \oplus aaabbb$ and $aaabaaabbbbb \notin L^*$.

(vi) Let L be a unary language over the alphabet $\{a\}$ such that $L_{even} = \{w \mid w$ contains an even number of a's$\}$. Clearly $L \in$ **SINS** as the sum of any two even numbers is even. Now consider the complement of L_{even}, $\overline{(L_{even})} = L_{odd} = \{w \mid w$ contains an odd number of a's$\}$. Then $aaa \in L_{odd}$ but $aaa \oplus aaa = \{aaaaaa\}$ which is not in L_{odd}.

(vii) Let $L = a^* \cup \{c\} \in$ **SINS**. Consider the homomorphism $h : \{a, b, c\}^* \rightarrow \{a, c\}^*$ given by: $h(a) = a$, $h(b) = \lambda$, $h(c) = c$. Then $h^{-1}(L) = \{a, b\}^* \cup b^*cb^*$ and $abb, cb \in h^{-1}(L)$ but $abb \oplus cb = abcbb$ which is not in $h^{-1}(L)$. ■

We recall that an anti-AFL is a language family that is not closed under any of the AFL operations: homomorphism, inverse homomorphism, intersection with regular languages, union, concatenation and Kleene Closure.

Corollary 3.1 **SINS** *is an anti-AFL.*

Proposition 3.6 *The family of sins-closed languages is closed under intersection.*

Proof. Let $L_1, L_2 \in$ **SINS**. We will show that $L_1 \cap L_2 \subseteq sins(L_1 \cap L_2)$.

Let $w \in L_1 \cap L_2$. By the definition of **SINS** we have that for all $v_1 \in L_1$, $v_1 \oplus w \subseteq L_1$. Likewise, for all $v_2 \in L_2$, $v_2 \oplus w \subseteq L_2$. It follows immediately that for all $v \in L_1 \cap L_2$, $v \oplus w \subseteq (L_1 \cap L_2)$ and thus $w \in sins(L_1 \cap L_2)$ and the proposition follows. ■

We consider now the closure properties of **SDEL** under standard operations.

Proposition 3.7 *The family of synchronized-deletion-closed languages is not closed under (i) union, (ii) catenation, (iii) homomorphism, (iv) intersection with a regular set, (v) Kleene closure (vi) complement or (vii) inverse homomorphism.*

Proof. (i) Consider $L_{ab} = \{w \mid |w|_a = |w|_b \geq 0\}, a^* \in$ **SDEL**. Clearly, $L_{ab} \cup a^*$ is not in **SDEL**.

(ii) The languages $L = \{a\}, L_{even} = \{w \mid w = a^n, n \text{ even}\}$ are in **SDEL**, however $L \cdot L_{even} = L_{odd}$ which is not in **SDEL**.

(iii) $L = (ba(aa)^*)$ is trivially in **SDEL**. Consider the homomorphism $h : \Sigma^* \to \Sigma^*$ given by $h(a) = a$, $h(b) = \lambda$. then $h(L) = a(aa)^* = L_{odd}$ which is not in **SDEL**.

(iv) The language L_{odd} is regular, and $L = a^*$ is in **SDEL**, however, $L_{odd} \cap a^* = L_{odd}$ which is not in **SDEL**.

(v) Let $\Sigma = \{a, b, c, d, e\}$ and $L = \{aaecc, bbedd, ccbbe\}$ which is trivially in **SDEL**. Then $aaeccbbedd, ccbbe \in L^*$ but $aaeccbbedd \ominus ccbbe = aaedd \notin L^*$.

(vi) L_{even} is in **SDEL** but $\overline{(L_{even})} = L_{odd}$ is not.

(vii) Let $L = \{ac, c\}$. Trivially, L is in **SDEL**. Now consider the homomorphism $h : \{a, b, c\}^* \to \{a, c\}^*$ given by $h(a) = a$, $h(b) = \lambda$, $h(c) = c$. Then $h^{-1}(L) = b^*ab^*cb^* \cup b^*cb^*$ and $abcb, cb \in h^{-1}(L)$. However, $abcb \ominus cb = \{ab\}$ which is not in $h^{-1}(L)$. ■

Corollary 3.2 **SDEL** *is an anti-AFL.*

Proposition 3.8 *The family of sdel-closed languages is closed under intersection.*

Proof. Symmetric to the proof of Proposition 3.6. ■

Since we have characterizations of sins- and sdel-closure in terms of synchronized bi-deletion, we briefly study families of languages closed under this operation.

Proposition 3.9 *Every intersection-closed full trio is closed under* $\boxminus$.

Proof. Let $\mathcal{L}$ be an intersection-closed full trio and let $L_1, L_2 \in \mathcal{L}$. Let M be an a-transducer that, on input w, guesses a partition of $w = xaz$ where $a \in \Sigma, x, z \in \Sigma^*$, and outputs xa followed by a new symbol c, then it outputs an arbitrary word $y \in \Sigma^*$, followed by ca, then z. Let L_2' be the language obtained from L_2 via M. Thus, every word of L_2' will be of the form $xacycaz$ where $xaz \in L_2$ and $y \in \Sigma^*$. Let M' be another a-transduction which guesses a partition of input $w = uavas$, where $a \in \Sigma, u, v, s \in \Sigma^*$ and outputs $uacvcas$. Let L_1' be the language obtained from L_1 via M'. Furthermore, let $L = L_1' \cap L_2'$. Every word of L is of the form $xacycaz$ where $xaz \in L_2, xayaz \in L_1$. Let L' be obtained from L by another a-transduction that erases everything until and including the first c, outputs the input until the next c, erases the c, outputs the next input symbol and erases everything after. Every word of L' is of the form ya where there exists $x, y, z \in \Sigma^*$ such that $xayaz \in L_1$ and $xaz \in L_2$.

Hence, $L' = L_1 \boxminus L_2$. ■

This leads to decidability questions regarding whether or not a language from a given language family is sins-closed or not.

Let **NCM** be the family of languages accepted by one-way, nondeterministic reversal-bounded multicounter automata [16] and let **DCM** by the deterministic variant. We will need some results from [16, 4], which are summarized in the following proposition:

Proposition 3.10 *1.* **NCM** *is effectively closed under intersection, is a full trio and has a decidable emptiness problem (given a machine M, is $L(M)$ empty?),*

2. **DCM** *is effectively closed under complementation.*

Proposition 3.11 *Let $\mathcal{L}_1, \mathcal{L}_2$ be language families, Σ a finite alphabet, $L \in \mathcal{L}_1, L \subseteq \Sigma^*$, satisfying the following conditions:*

1. *$\mathcal{L}_1$ is effectively contained in $\mathcal{L}_2$ and is effectively closed under complementation,*
2. *$\mathcal{L}_2$ is effectively closed under intersection and the full trio operations with a decidable emptiness problem.*

Then it is decidable whether L is sins-closed over Σ and whether L is sdel-closed over Σ.

Proof. It is enough to decide whether there exists $w \in L$ such that $w \notin sins(L)$ (respectively $sdel(L)$). Further, $w \notin sins(L)$ if and only if $w \in \overline{L} \boxminus L$ (respectively $L \boxminus \overline{L}$), by Proposition 3.2 (respectively 3.3). Since $\mathcal{L}_1$ is closed under complementation, $\overline{L} \in \mathcal{L}_1 \subseteq \mathcal{L}_2$. By Proposition 3.9, $\overline{L} \boxminus L \in \mathcal{L}_2$ (respectively $L \boxminus \overline{L} \in \mathcal{L}_2$). Further, since $\mathcal{L}_2$ is intersection-closed, $(\overline{L} \boxminus L) \cap L \in \mathcal{L}_2$ (respectively $(L \boxminus \overline{L}) \cap L \in \mathcal{L}_2$). Thus, this set is empty if and only if L is sins-closed (respectively sdel-closed).

Hence, by the decidability of emptiness for $\mathcal{L}_2$, it is decidable whether an arbitrary language in $\mathcal{L}_1$ is sins-closed or sdel-closed. ■

Combining Propositions 3.10 and 3.11, we obtain:

Corollary 3.3 *Let Σ be an alphabet, $L \in$* **DCM**$, L \subseteq \Sigma^*$. *Then it is decidable whether L is sins-closed over Σ and whether L is sdel-closed over Σ.*

The result above also holds for the family of regular languages. Also, it is known that **NCM** is effectively equal to the finite-crossing nondeterministic reversal-bounded multicounter languages (finite-crossing refers to a two-way input tape where there is a bound on the number of switches between moving left and right on the input tape) [12]. It is also known that the family of finite-crossing deterministic reversal-bounded multicounter languages is closed under complement [16], so the result above is also true for this language family as well.

The question arises of whether decidability remains true when nondeterminism is used. We will show that even for one-way, nondeterministic 1-reversal bounded single counter languages, it is undecidable whether an arbitrary language is *sins*-closed. Let $\mathbf{NCM}(\mathbf{1},\mathbf{1})$ denote this language family.

A valid computation [15] of a Turing Machine M is a string $cx_1cx_2c\cdots cx_mc$ where x_i is an instantaneous description of M for all i, x_1 is an initial instantaneous description, x_m is an instantaneous description with final state and x_{i+1} follows from x_i by one move of M, for all i. An invalid computation of M is any string that is not of this form. It is known that the language of invalid computations of M can be accepted by a one-way, nondeterministic one-reversal bounded single counter machine [1].

Proposition 3.12 *Let Σ be a finite alphabet, $L \in \mathbf{NCM}(\mathbf{1},\mathbf{1}), L \subseteq \Sigma^*$. Then it is undecidable whether L is sins-closed over Σ and whether L is sdel-closed over Σ.*

Proof. Let M be a Turing Machine and let $L \in \mathbf{NCM}(\mathbf{1},\mathbf{1})$ be the language of invalid computations of M. Thus, $\overline{L}$ is the language of valid computations of M. As noted above, L is sins-closed (respectively sdel-closed) if and only if $L \cap (\overline{L} \boxminus L) = \emptyset$ (respectively $L \cap (L \boxminus \overline{L}) = \emptyset$).

Assume that $\overline{L}$ is empty. Then clearly $\overline{L} \boxminus L$ is empty and so $L \cap (\overline{L} \boxminus L)$ is also. Similarly, $L \cap (L \boxminus \overline{L})$ is empty.

Assume that there exists $w \in \overline{L}$. Then the string c is in L since c is an invalid computation (has to enter at least one state). Also, $\{w\} \boxminus \{c\}$ is non-empty since $c^{-1}w \in \{w\} \boxminus \{c\}$ and it is also an invalid computation. Thus, $L \cap (\overline{L} \boxminus L)$ is non-empty. Similarly, $wc \in L$ since it is an invalid computation always. Let $w = w_1c$. But then $c \in w_1cc \boxminus w_1c$. Consequently, $L \boxminus \overline{L}$ is nonempty and $L \cap (L \boxminus \overline{L})$ is nonempty since c is an invalid computation.

Thus, $L \cap (\overline{L} \boxminus L) = \emptyset$ and $L \cap (L \boxminus \overline{L}) = \emptyset$ if and only if $\overline{L}$ is empty if and only if the language of valid computations of M is empty, which is undecidable.

Hence, it is undecidable whether L is sins-closed and whether L is sdel-closed. ■

Corollary 3.4 *Let Σ be an alphabet. It is undecidable whether a linear or context-free language over Σ is sins-closed over Σ. Furthermore, it is undecidable whether a linear or context-free language over Σ is sdel-closed over Σ.*

The question remains of whether decidability is still true when we remove the reversal-bounded condition on deterministic multicounter automata. We will see that this is not the case even when there is only one counter.

Let $\mathbf{DCM}(\mathbf{1})$ be the family of languages accepted by one-way deterministic single counter automata (no reversal bound).

Proposition 3.13 *Let $\mathcal{L}$ be a language family, Σ an alphabet, $L \in \mathcal{L}, L \subseteq \Sigma^*$ satisfying the following properties:*

1. *if $L_1, L_2 \in \mathcal{L}$, R a regular language, c, d new symbols, then $c(L_1d)^+ \cup (L_2d)^+, L_1 \cup R \in \mathcal{L}$, both effective.*

2. *$\mathcal{L}$ has an undecidable inclusion problem (given $L_1, L_2 \in \mathcal{L}$, is $L_1 \subseteq L_2$?).*

Then it is undecidable whether $L \in \mathcal{L}$ is sins-closed over Σ.

Proof. Let $L_1, L_2 \in \mathcal{L}$ and c, d be new symbols. Let $L_1' = c(L_1d)^+$ and $L_2' = (L_2d)^+$. Let $L_3 = \{a_1 \cdots a_n \mid c \in \text{alph}(a_2 \cdots a_n)\}$. It is obvious that L_3 is a regular language and by the assumptions, $L = L_1' \cup L_2' \cup L_3 \in \mathcal{L}$. We will show that L is sins-closed if and only if $L_2 \subseteq L_1$. Let $x \in L$.

First, assume that $x \in L_1'$. Then $\alpha \oplus x \subseteq L_3 \subseteq L$ for all $\alpha \in L_1'$. Also, $\alpha \oplus x \subseteq L_3 \subseteq L$ for all $\alpha \in L_3$. Further, $\alpha \oplus x = \{udvdy \mid \alpha = udy, x = vd\}$, d as above, for all $\alpha \in L_2'$ where the first letter of v is c. Thus, $\alpha \oplus x \subseteq L_3$ for all $\alpha \in L_2'$. Therefore, $L_1' \subseteq sins(L)$.

Second, assume that $x \in L_3$. Then $\alpha \oplus x \subseteq L_3 \subseteq L$ for all $\alpha \in L_1'$. Also, $\alpha \oplus x \subseteq L_3 \subseteq L$ for all $\alpha \in L_2'$. Further, $\alpha \oplus x \subseteq L_3 \subseteq L$ for all $\alpha \in L_3$. Hence, $L_3 \subseteq sins(L)$.

Last, assume that $x \in L_2'$. Then $\alpha \oplus x \subseteq L_2 \subseteq L$ for all $\alpha \in L_2'$. Also, $\alpha \oplus x \subseteq L_3 \subseteq L$ for all $\alpha \in L_3$. Further, $\alpha \oplus x \subseteq L$ if and only if $\alpha \oplus x \subseteq L_1'$ for all $\alpha \in L_1'$.

Thus, L is sins-closed if and only if $\alpha \oplus x \subseteq L_1'$ for all $\alpha \in L_1', x \in L_2'$. Assume that $\alpha \oplus x \subseteq L_1'$ for all $\alpha \in L_1', x \in L_2'$. Then it must be the case that $L_2 \subseteq L_1$. Assume that $L_2 \subseteq L_1$. Consequently, $\alpha \oplus x \subseteq L_1'$ for all $\alpha \in L_1', x \in L_2'$.

Hence, L is sins-closed if and only if $L_2 \subseteq L_1$, which is an undecidable problem. ■

Corollary 3.5 *Let $L \in \mathbf{DCM}(\mathbf{1})$, Σ an alphabet, $L \subseteq \Sigma^*$. Then it is undecidable whether L is sins-closed over Σ. In addition, it is undecidable whether a deterministic context-free language is sins-closed over Σ.*

Proof. It is known that the inclusion problem is undecidable for the language family $\mathbf{DCM}(\mathbf{1})$ [16]. Moreover, the languages L_1' and L_2' in the proof above are in $\mathbf{DCM}(\mathbf{1})$ since $\mathbf{DCM}(\mathbf{1})$ is closed under marked[2] + and concatenation on the left by a new symbol [14]. Further $L_1' \cup L_2' \in \mathbf{DCM}(\mathbf{1})$ since if the first letter is c then do the first automaton, otherwise do the second. It is also known that $\mathbf{DCM}(\mathbf{1})$ is closed under union with regular languages (This follows from the facts that $\mathbf{DCM}(\mathbf{1})$ is closed under complement, [14], $\mathbf{DCM}(\mathbf{1})$ is closed under intersection with regular languages

[2]The marked + of a language L is $(Ld)^+$ where d is a new symbol.

since the standard proof [15] of closure of deterministic context-free languages under intersection with regular languages does not increase the size of the pushdown alphabet and the fact that $L \cup R = \overline{(\overline{L} \cap \overline{R})}$ where R is a regular language). Thus, the corollary holds. ■

Similar to the results for **SINS**, we obtain undecidability with deterministic single counter languages for **SDEL**.

Proposition 3.14 *Let $\mathcal{L}$ be a language family, Σ a finite alphabet, $L \in \mathcal{L}, L \subseteq \Sigma^*$ satisfying the following properties:*

1. *if $L_1, L_2 \in \mathcal{L}, c, d$ new symbols, then $dcL_1d \cup cL_2d \in \mathcal{L}$ and $\overline{L_1} \in \mathcal{L}$, both effective,*
2. *$\mathcal{L}$ has an undecidable inclusion problem.*

Then it is undecidable whether L is sdel-closed over Σ.

Proof. Let $L_1, L_2 \in \mathcal{L}$ and c, d be new symbols. Let $L_1' = dc\overline{L_1}d$, $L_2' = cL_2d$ and let $L = L_1 \cup L_2 \in \mathcal{L}$, by assumption.

We will show that L is sdel-closed if and only if $L_2 \subseteq L_1$. Let $x \in L$.

First, assume that $x \in L_1'$. Then $\alpha \ominus x = \emptyset \subseteq L$ for all $\alpha \in L_1'$. Further, $\alpha \ominus x = \emptyset \subseteq L$ for all $\alpha \in L_2'$. Hence, $L_1' \subseteq sdel(L)$.

Then, assume that $x \in L_2'$. So, $\alpha \ominus x = \emptyset \subseteq L$ for all $\alpha \in L_2'$. Further, $\alpha \ominus x$ is always equal to either $\{d\}$ or the emptyset for each $\alpha \in L_1'$. Assume that $\{d\} = \alpha \ominus x$. Thus, $\alpha = dcvd$ and $x = cvd$ for some $v \in \overline{L_1} \cap L_2$. Hence, $L_2 \not\subseteq L_1$. Assume that $\alpha \ominus x = \emptyset$. Thus, $v \notin \overline{L_1} \cap L_2$ where $\alpha = dcvd, x = cvd$. So, if $L_1' \ominus L_2' = \emptyset$ then $\overline{L_1} \cap L_2 = \emptyset$ and $L_2 \subseteq L_1$. Of course, $\emptyset \subseteq L$ and $\{d\} \not\subseteq L$.

Hence, L is sdel-closed if and only if $L_2 \subseteq L_1$, which is an undecidable problem. ■

In the proof above, it is immediate that $L_1', L_2' \in \mathbf{DCM}(\mathbf{1})$ since $\mathbf{DCM}(\mathbf{1})$ is closed under concatenation on the left and right by a new symbol and also complementation [14]. Also, $L \in \mathbf{DCM}(\mathbf{1})$ since if the first letter is d then do M_1' otherwise do M_2'.

Corollary 3.6 *Let Σ be an alphabet, $L \in \mathbf{DCM}(\mathbf{1}), L \subseteq \Sigma^*$. Then it is undecidable whether L is sdel-closed over Σ. Moreover, it is undecidable whether a deterministic context-free language is sdel-closed over Σ.*

4 Hi, dlad and ld

There are also three unary operations, hi, dlad and ld from [7, 8, 9], inspired by intramolecular DNA recombination. We now recall the definitions of the hi and dlad operations.

Definition 4.1 *Let $w \in \Sigma^*$.*

1. *The hairpin inverse of w, denoted by $hi(w)$ is defined as*

$$hi(w) = \{xpy^R p^R z \mid w = xpyp^R z \text{ and } x, y, z \in \Sigma^*, p \in \Sigma^+\}.$$

2. *The double loop with alternating direct pointers operation on w, denoted by $dlad(w)$ is defined as*

$$dlad(w) = \{xp\alpha qyp\beta qz \mid w = xp\beta qyp\alpha qz, x, y, z, \alpha, \beta \in \Sigma^*, p, q \in \Sigma^+\}.$$

The operations are then extended to unary operations on languages in the natural way.

We recall the following lemma from [3, 4].

Lemma 4.1 *Let $w \in \Sigma^*$. Then*

1. $hi(w) = \{xay^R az \mid w = xayaz, a \in \Sigma, x, y, z \in \Sigma^*\}$,
2. $dlad(w) = \{xa\alpha bya\beta bz \mid w = xa\beta bya\alpha bz, x, y, z, \alpha, \beta \in \Sigma^*, a, b \in \Sigma\}$.

We call the letters a and b above the pointers.

Based on Lemma 4.1, we shall always assume without loss of generality that the pointers for the hi and dlad operators are of length one.

We find it convenient for Proposition 4.1 to first define the following variant of the hairpin inversion operator. This variant forces the pointer to be a particular letter.

Definition 4.2 *Let Σ be an alphabet, $w \in \Sigma^*$ and let $a \in \Sigma$. The a-projected hairpin inverse of w, denoted by $hi_a(w)$ is defined as $hi_a(w) = \{xay^R az \mid w = xayaz, x, y, z \in \Sigma^*\}$.*

We extend this definition to languages similar to above.

Lemma 4.2 *Let $\mathcal{L}$ be a trio closed under hi. Then for each $L \in \mathcal{L}$ over Σ, and for each $a \in \Sigma$, $hi_a(L) \in \mathcal{L}$.*

Proof. Let $\$_1, \$_2, \$_3, \$_4$ be new symbols. We construct an nondeterministic gsm M that outputs the input until it reaches an arbitrary occurrence of the letter a. Then, M outputs $\$_1 a \$_2$. Then, M continues to output the input until another arbitrary occurrence of the letter a, where M outputs $\$_3 a \$_4$ and continues to output the input. Let h be a homomorphism that erases all \$ symbols and leaves all others fixed. Consider $L' = h(hi(M(L)) \cap \Sigma^* \$_1 a \$_3 \Sigma^* \$_2 a \$_4 \Sigma^*)$. It is clear that $L' = hi_a(L)$. Furthermore, every trio is closed under λ-free nondeterministic gsm mappings, limited erasing homomorphisms that do not introduce the empty word and intersection with regular languages. ■

This shows that every *hi*-closed trio is also closed under the projected *hi* operator, for each letter. This will significantly simplify the proof of Proposition 4.1.

Next, we see that for all trios, closure under *dlad* is a consequence of closure under *hi*.

Proposition 4.1 *Let $\mathcal{L}$ be a trio closed under hi. Then $\mathcal{L}$ is closed under dlad.*

Proof. Let $L \in \mathcal{L}$ over Σ. Let M be an λ-free a-transducer $M = (Q, \Sigma, \Sigma \cup \Delta, \delta, q_0, \{q\})$ where $\Delta = \{\$, \$_1, \$_2, \$_3\}$, all new symbols, Σ is the input alphabet, $\Sigma \cup \Delta$ is the output alphabet, $Q = \{q_0, q\} \cup \{q_a, q_{ab}, q'_a \mid a, b \in \Sigma\}$ is the set of states, q is the final state and the transitions are defined by, δ=$\{(q_0, a, a, q_0)$, $(q_0, a, \$\$_1 a, q_a)$, (q_a, b, b, q_a), $(q_a, b, b\$_1\$_2, q_{ab})$, (q_{ab}, c, c, q_{ab}), $(q_{ab}, a, \$_2\$_3 a, q'_b)$, (q'_b, c, c, q'_b) , $(q'_b, b, b\$_3\$, q)$, $(q, c, c, q)\}$ for all $a, b, c \in \Sigma$. Intuitively, M outputs the input until it reaches an arbitrary input letter p, where M outputs $\$\$_1 p$ and continues outputting the input. Then at an arbitrary letter q, M outputs $q\$_1\$_2$ and continues to output the input. Then, at an arbitrary occurrence of the letter p (the same letter as above which we keep track of using the subscript on the states), M outputs $\$_2\$_3 p$ and continues to output the input. Then, at an arbitrary occurrence of q (also as above), M outputs $q\$_3\$$. Let $L' = M(L)$. Every word of L' is of the form

$$x\$\$_1 p \beta q \$_1 \$_2 y \$_2 \$_3 p \alpha q \$_3 \$ z \tag{1}$$

with $x, y, z, \alpha, \beta \in \Sigma^*, p, q \in \Sigma$. Consider $L'' = hi_{\$_3}(hi_{\$_2}(hi_{\$_1}(hi_{\$}(L'))))$. This sequence of operations first flips the entire subword between the first and last pointer, then the three subwords between each pointer. Thus, the word of L'' corresponding to the word in (1) is of the form,

$$x\$\$_3 p \alpha q \$_3 \$_2 y \$_2 \$_1 p \beta q \$_1 \$ z.$$

Let h be a homomorphism that erases all \$ letters and leaves all others unchanged. Clearly, $h(L'') = dlad(L)$. Furthermore, every hi-closed trio is closed under the projected *hi* operators, λ-free gsm mappings and limited erasing homomorphisms that do not introduce the empty word.

Hence, every *hi*-closed trio is closed under dlad. ■

This serves to simplify the closure of some language families under *dlad* [4]. Furthermore, this result combined with the proof that every full trio is closed under the ld bio-operator, implies that every hi-closed full trio is closed under all three bio-operators inspired by intramolecular DNA recombination [7, 8, 9].

Corollary 4.1 *Every hi-closed full trio is closed under hi, dlad and ld.*

This shows the power and importance of the hairpin inversion operation.

5 Hi-closed languages

The previous section demonstrates the importance of the hairpin inversion operation. In this section, we will study languages that are closed under hairpin inversion.

We say that a language $L \subseteq \Sigma^*$ is *hi*-closed if $hi(L) \subseteq L$.

Example 5.1

1. $hi(a^+) = aa^+$
2. *Let* $L_{ab} = \{w \mid |w|_a = |w|_b \geq 0\}$. *Then* $hi(L_{ab}) = \{w \mid |w|_a = |w|_b > 1\}$.
3. $hi(\{a^n b^n \mid n \geq 0\}) = \{a^n b^n \mid n \geq 2\}$
4. $hi(\{abca\}) = \{acba\}$

We see that numbers (1), (2) and (3) of example 3.1 are *hi*-closed.

We denote the family of *hi*-closed languages by **HI**. By the definition of *hi*-closed languages, it is clear that if $L \in \mathbf{HI}, w \in L$ then $hi(w) \subseteq L$. We note that $hi(hi(L)) \subseteq L$. Also, if $w \in L$ and $w = uavax$ for some $a \in \Sigma$, then $w \in hi(hi(w))$. Therefore, for all $w \in L$ such that $|w|_a \geq 2$ for some $a \in \Sigma$, $w \in hi(L)$. Let R_Σ be the finite set of all words over Σ such that $w \in R_\Sigma$ implies that $|w|_a < 2$ for all $a \in \Sigma$. Hence, if L is hi-closed, it follows that $L - R_\Sigma = hi(L)$.

By the effective closure of regular languages under *hi* [3] and the decidability of the inclusion problem, we obtain:

Proposition 5.1 *Let L be a regular language. Then it is decidable whether $L \in \mathbf{HI}$.*

This could also be shown by taking the set difference of L with the finite language R_Σ and testing for equality with $hi(L)$.

We would like to consider similar questions for more general families of languages. We first will need to develop some notation.

Definition 5.1 *Let $w \in \Sigma^*, \Sigma = \{a_1, \ldots, a_k\}$. We denote by*

$$\pi(w) = \begin{cases} \lambda & \text{if } w = \lambda, \\ a_{i_1} \cdots a_{i_m} & \text{otherwise if } w = a_{i_1}^{j_1} \cdots a_{i_m}^{j_m}, i_l \neq i_{l+1} \text{ for} \\ & \text{all } 1 \leq l \leq m-1 \text{ and } j_i > 0 \text{ for all } 1 \leq i \leq m. \end{cases}$$

We define an equivalence relation $\equiv \subseteq \Sigma^* \times \Sigma^*$ by $w \equiv w'$ if and only if $\pi(w) = \pi(w')$ and $\Psi(w) = \Psi(w')$. We denote by $[w]$ the equivalence class of w.

Proposition 5.2 *Let $\Sigma = \{a, b\}, L \subseteq \Sigma^*$. Then L is a (potentially infinite) union of equivalence classes of $\equiv$ if and only if $L \in$ **HI**.*

Proof. "$\Rightarrow$" Suppose that L is a union of equivalence classes of $\equiv$. Let $w \in hi(L)$. Thus $w = uav^Ray$ where $w' = uavay \in L$. Notice that $\Psi(ava) = \Psi(av^Ra)$. Furthermore, since L is over a binary alphabet, $\pi(ava) = \pi(av^Ra)$. Thus, $\Psi(w) = \Psi(w')$ and $\pi(w) = \pi(w')$. Consequently, $w \equiv w'$, $w \in L$ and $hi(L) \subseteq L$.

"$\Leftarrow$" Suppose $L \in$ **HI**. Therefore , $hi(L) \subseteq L$. Let $w \in L$. Consider $v \in [w]$. Thus, $\Psi(v) = \Psi(w)$ and $\pi(v) = \pi(w)$. We will show that $v \in L$. If $w = v = \lambda$ then $v \in L$. Assume then that $w, v \in \Sigma^+$. First, we will show that if the maximal common prefix [3] of v and w is of size n, where $0 \leq n < |w|$, then we can construct a word $w' \in L$ such that $[v] = [w] = [w']$ and the maximal common prefix of v and w' is of size greater than n.

We note that the maximal common prefix of w and v must be at least one since $\pi(w) = \pi(v)$. If the maximal common prefix of w and v is of length $|w|$, then $w = v \in L$. Assume that the maximal common prefix of v and w is of length $n, 1 \leq n < |w|$. Let the $n+1^{\text{st}}$ position of w be c, the $n+1^{\text{st}}$ position of v be d (with $c \neq d$) and the common n^{th} position be e, where $c, d, e \in \{a, b\}$. Indeed, $w = xecx_1, v = xedy_1, x, x_1, y_1 \in \{a, b\}^*, e, c, d \in \{a, b\}, c \neq d$.

Assume that $e = c$, therefore $w = xccx_1, v = xcdy_1$. Since $|w|_d = |v|_d$ and Σ is binary, it follows that $w = xccx_2dx_3$, with $x_2 \in c^*, x_3 \in \Sigma^*$. Furthermore, since $|w|_c = |v|_c$ and Σ is binary, it follows that $v = xcy_2dcy_3$, with $y_2 \in d^*, y_3 \in \Sigma^*$. However, since $\pi(w) = \pi(v)$, (in particular since $\pi(cy_2dc) = \pi(cdc)$), we can rewrite $w = xccx_2dx_4cx_5, x_2 \in c^*, x_4 \in d^*, x_5 \in \Sigma^*$. Because L is *hi*-closed, $w' = xcx_4^Rdx_2^Rccx_5 \in L, [w'] = [w]$ since Σ is binary and the maximal common prefix of v and w' is of length greater than n.

Assume that $e = d$, therefore $w = xdcx_1, v = xddy_1$. We say that a word α is an infix of β, denoted $\alpha \leq_{\text{i}} \beta$ if there exists words $\alpha_1, \alpha_2 \in \Sigma^*$ such that $\beta = \alpha_1\alpha\alpha_2$.

[3] The maximal common prefix of $v, w \in \Sigma^*$, is the longest, unique word $u \in \Sigma^*$ such that $v = uv_1, w = uv_2$ for some $v_1, v_2 \in \Sigma^*$.

Claim 5.1 $dd \leq_{\mathrm{i}} x_1$.

Proof. Assume otherwise. We know that both $\pi(dcx_1) = \pi(ddy_1)$ and that $\Psi(dcx_1) = \Psi(ddy_1)$ hold. By the assumption, $dcx_1 \in (dc^+)^m c^*$ for some $m \geq 1$. Since, $\pi(dcx_1) = \pi(ddy_1)$, it must be the case that $ddy_1 \in (d^+c^+)^m c^*$. However, $|ddy_1|_d > m$ since the first two letters are dd and $|dcx_1|_d = m$, a contradiction. ■

Consequently, $w = xdcx_2ddx_3$ for some $x_2, x_3 \in \Sigma^*$. Furthermore, $w' = xddx_2^R cdx_3 \in L$ for some $x_2, x_3 \in \Sigma^*$ since L is hi-closed, it is clear that $[w] = [w']$ since Σ is binary and the maximal common prefix of w' and v is of length at least $n + 1$.

Therefore, if the maximal common prefix is of length $|w|$, then $w = v \in L$. Assume that the maximal common prefix is of size $1 \leq n < |w|$. Then we can construct a word $w' \in L$ such that $[v] = [w] = [w']$ and the maximal common prefix of w' and v is of length greater than n. Therefore, by way of induction, we can construct a word w^i with maximal common prefix of size $|v|$ with v such that $[v] = [w] = [w^i]$ and $w^i \in L$. Thus, $v = w^i \in L$ and $[v] \subseteq L$.

Hence L is a union of equivalence classes of $\equiv$ if and only if L is hi-closed. ■

Thus, $L \in \mathbf{HI}, L \subseteq \{a, b\}^*$ is a (potentially infinite) union of equivalence classes of $\equiv$. However, each equivalence class has a relatively simple structure. Indeed, for any equivalence class $[w]$ where $w \in L$, we can uniquely represent $[w]$ by the number of a's of any word in $[w]$, the number of b's of any word in $[w]$, the number n where for every word $v \in [w]$, $v \in (c^+d^+)^n c^*, c, d \in \{a, b\}, c \neq d$, the starting letter c, and whether the trailing c^* is empty or not.

Let **NPCM** be the family of languages accepted by machines defined by a nondeterministic pushdown augmented by k-reversal bounded counters, for any k, and a one-way input tape. It is known that the emptiness problem is decidable for **NPCM** [16, 13]. Let **DPCM** be the family of languages accepted by the deterministic variant. It is known that **DPCM** is closed under complement [16]. Using ideas in [13], we can prove the following proposition:

Proposition 5.3 *Let $\Sigma = \{a, b\}, L \in \mathbf{DPCM}, L \subseteq \Sigma^*$. Then it is decidable whether $L \in \mathbf{HI}$.*

Proof. Let $L = L(M) \in \mathbf{DPCM}$ for some machine M. We will construct a machine $M' \in \mathbf{NPCM}$ such that $L(M') = \emptyset$ if and only if $L \in \mathbf{HI}$. Let \$ be a new symbol. Indeed, M' will accept all words $w\$w'$ where $w \in L, w' \in \overline{L}$ and $[w] = [w']$. Let $w\$w'$ be the input. By Proposition 5.2, $L(M') = \emptyset$ if and only if $L \in \mathbf{HI}$.

Construct M' such that M' simulates M on w and in parallel, records the first letter c and the last letter e in the finite control and n_1, n_2, n_3 in three new 1-reversal-bounded counters, where $n_1 = |w|_a, n_2 = |w|_b$ and n_3 is such that $w \in (c^+d^+)^{n_3}c^*$ for

some $c, d \in \{a, b\}, c \neq d$. If $w \in L$, then, in parallel, verify both that $w' \in \overline{L}$ and that $[w'] = [w]$, ie. $|w'|_a = n_1, |w'|_b = n_2$, the first letter is c, the last letter is e, and that n_3 is such that $w' \in (c^+d^+)^{n_3}c^*$. Thus $w\$w' \in L(M')$ if and only if $w \in L$, $w' \notin L$ such that $[w] = [w']$.

Hence it is decidable whether $L \in \mathbf{HI}$. ■

Corollary 5.1 *Let $\Sigma = \{a, b\}$, L a deterministic context-free language, $L \subseteq \Sigma^*$. Then it is decidable whether $L \in \mathbf{HI}$.*

We, as yet, have been unable to prove this for alphabets of arbitrary size.

The question arises of whether decidability remains true when nondeterminism is used. We will show that even for one-way, nondeterministic 1-reversal bounded single counter languages, it is undecidable whether an arbitrary language is *hi*-closed. Let $\mathbf{NCM}(\mathbf{1}, \mathbf{1})$ denote this language family.

To see the undecidability of *hi*-closure for this family, we observe that it is undecidable whether the language accepted by an arbitrary Turing Machine M that makes at least two moves and has distinct initial and final states is empty. Furthermore, it is known that the language of invalid computations of M can be accepted by a one-way, nondeterministic one-reversal bounded counter automaton [1]. We will use the following lemma:

Lemma 5.1 *Let $L \subseteq \Sigma^*$ and let $L \in \mathbf{HI}$. Then $w = cw_1cw_2cw_3c \in L$ if and only if $w' = cw_3cw_2cw_1c \in L$.*

Proof. Assume $w = cw_1cw_2cw_3c \in L$. Then $cw_3^Rcw_2^Rcw_1^Rc \in L$ by reversing the word between the two outermost c's and $cw_3cw_2cw_1c \in L$ by then reversing the word between the first two c's, the second and third c and then the third and fourth c. The converse is similar. ■

Lemma 5.2 *Let M be a Turing Machine that has disjoint start and final states and makes at least two moves before accepting any word. Let L_M be the language of invalid computations of M. Then L_M is hi-closed if and only if $L_M = \Sigma^*$.*

Proof. "⇒" Assume that L_M is *hi*-closed and $L_M \neq \Sigma^*$. Thus there exists a valid computation $cx_1cx_2c \cdots cx_mc$. However, $cx_mcx_2c \cdots cx_1c \in L_M$ since $m > 2$ and the final and initial states are disjoint. But since L_M is *hi*-closed, $cx_1cx_2c \cdots cx_mc \in L_M$, by Lemma 5.1, a contradiction.

"⇐" It is immediate that Σ^* is hi-closed. ■

This, in conjunction with the fact that it is undecidable whether a Turing Machine that has disjoint initial states and final states that must make at least two moves on an accepting computation is empty, we obtain:

Proposition 5.4 *Let $L \in \mathbf{NCM}(\mathbf{1},\mathbf{1})$. Then it is undecidable whether $L \in \mathbf{HI}$.*

As corollary, it is immediate that:

Corollary 5.2 *Let L be a context-free or linear language. Then it is undecidable whether $L \in \mathbf{HI}$.*

We next sum up the closure properties of the family **HI**.

Proposition 5.5 *The family of hi-closed languages is not closed under (i) intersection with regular languages, (ii) intersection with finite languages, (iii) homomorphism, (iv) λ-free homomorphism, (v) concatenation, (vi) $*$, (vii) $+$ or (viii) inverse homomorphism.*

Proof. (i),(ii) Let $L_1 = \{abaa, aaba\} \in \mathbf{HI}$. However, $L_1 \cap \{abaa\} = \{abaa\} \notin \mathbf{HI}$. Hence, **HI** is not closed under intersection with regular or finite sets.

(iii),(iv) Let $L_2 = \{ccabd\} \in \mathbf{HI}$. Let h be a homomorphism that maps the letter d to c and leaves all others unchanged. But, $h(L) = \{ccabc\} \notin \mathbf{HI}$. Hence, **HI** is not closed under homomorphism or λ-free homomorphism.

(v) Let $L_3 = \{ccab\}, L_4 = \{cc\} \in \mathbf{HI}$. But, $L_3L_4 = \{ccabcc\} \notin \mathbf{HI}$. Hence **HI** is not closed under concatenation.

(vi),(vii) Let $L_5 = L_3^*$. Then $ccabccab \in L_5$, but $ccbaccab \notin L_5$. Hence **HI** is not closed under $*$. Similarly, **HI** is not closed under $+$.

(viii) Let $L_6 = \{bcdebc, bcbedc, bedcbc\}$. Let h be a homomorphism from the set $\{a,d,e\}^*$ to $\{b,c,d,e\}^*$ that maps a to bc, d to d and e to e. Then $h^{-1}(L_6) = \{adea\} \notin \mathbf{HI}$. Hence, **HI** is not closed under inverse homomorphism. ■

However, we see that the family is closed under three standard operations.

Proposition 5.6 HI *is closed under union, complementation and intersection.*

Proof. Let $L \in \mathbf{HI}$. Therefore, if $w \in L$, then $hi(w) \subseteq L$. Let $v \in \overline{L}$. Assume that $hi(v) \not\subseteq \overline{L}$. Thus, there exists $u \in hi(v)$ such that $u \in L$. So $hi(u) \subseteq L$. However, $v \in hi(u)$ since $u \in hi(v)$. Thus, $v \in L$, a contradiction. Hence, **HI** is closed under complementation.

Let $L_1, L_2 \in \mathbf{HI}$, $L = L_1 \cup L_2$ and $w \in L$. Then either $w \in L_1$ or $w \in L_2$. If $w \in L_1$ then $hi(w) \subseteq L_1$ and if $w \in L_2$ then $hi(w) \subseteq L_2$. Thus, $hi(w) \subseteq L$. Hence, **HI** is closed under union.

Closure under intersection follows from both closure under union and complement. ■

6 Conclusions

In this paper we have studied the properties of, and relationships between, families of languages defined by closure under the synchronized insertion, synchronized deletion, and hairpin inversion operations.

We have shown that while the family of synchronized-insertion-closed languages is closed under intersection, it is not closed under union, catenation, homomorphism, inverse homomorphism, intersection with a regular sets, Kleene closure or complement and is thus an anti-AFL. We have also shown that it is decidable if a language defined by a one-way deterministic reversal-bounded multicounter machine is a synchronized-insertion-closed language. The same question is undecidable for languages defined by one-way non-deterministic 1-reversal-bounded single counter machines and languages defined by one-way deterministic single counter automata.

Similarly, we showed that the family of synchronized-deletion-closed languages is an intersection-closed anti-AFL. Furthermore, it is decidable if a given deterministic reversal-bounded multicounter language is a synchronized-deletion-closed language while the same question is undecidable for one-way non-deterministic 1-reversal-bounded single counter languages and one-way deterministic single counter languages.

Families of languages defined by closure under the hairpin-inversion operation were shown to be closed under union, intersection and complementation but not under intersection with regular languages, homomorphism, inverse homomorphism, concatenation or Kleene closure. Moreover, it is decidable if an arbitrary regular language is a hairpin-inversion-closed language while the same question is undecidable for one-way non-deterministic 1-reversal bounded single counter languages. We further showed that it is decidable if a language defined by a deterministic pushdown machine augmented with multiple reversal-bounded counters over a binary alphabet is a hairpin-inversion-closed language. This question remains open over alphabets of arbitrary size.

We additionally demonstrated the central importance of the hairpin inversion operation with the result that, for any trio, closure under hairpin inversion implies closure under the dlad operation.

By considering families of languages that are defined by closure under a biological gene descrambling operation, we gain insight into the structure of "maximally descrambled" genomes. That is, a language closed under a particular operation represents a fixed-point with respect to iterated application of that operation. Information of this sort can carry a great deal of biological relevance. In this case, if we view the descrambling process as a technique to intentionally induce evolution through genetic mutation, then the language families studied in this paper represent "dead end" genomes in which no further evolution is possible using the chosen operation.

Of further interest are families of languages defined by closure under multiple op-

erations, the ld and dlad operations, as well as the new template-guided descrambling model introduced in [21] and studied formally in [6]. It is hoped that further study of the theoretical properties of these models will lead us to new insights in the genetics of stichotrichous ciliates.

References

[1] B. Baker, R. Book, Reversal-bounded multipushdown machines, *Journal of Computer and System Sciences* 8 (1974) 315–332.

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

[3] M. Daley, O. Ibarra, L. Kari, Closure properties and decision questions concerning some language classes under ciliate bio-operations, *Theoretical Computer Science*, 306 (2003) 19–38.

[4] M. Daley, O. Ibarra, L. Kari, I. McQuillan, K. Nakano, The ld and dlad Bio-Operations on Formal Languages, *Journal of Automata, Languages and Combinatorics*, 8 (2003) 477–498.

[5] Mark Daley and Lila Kari. Some properties of ciliate bio-operations. In M. Ito and M. Toyama, editors, *Developments in Language Theory 2002, Sixth International Conference (DLT 2002), Kyoto, Japan*, number 2450 in Lecture Notes in Computer Science, pp. 116–127. Springer-Verlag, 2003.

[6] M. Daley, I. McQuillan, Template-guided DNA recombination, in: E. Csuhaj-Varjú, C. Kintala, D. Wotschke, G. Vaszil (Eds.), *Fifth International Workshop, Descriptional Complexity of Formal Systems, Budapest, Hungary, July 12-14, 2003*, Proceedings, MTA SZTAKI, 2003, pp. 235–244.

[7] A. Ehrenfeucht, T. Harju, I. Petre, G. Rozenberg, Patterns of micronuclear genes in ciliates. In N. Jonoska, N. Seeman, editors, *Seventh International Meeting on DNA-based Computers (DNA 7)*, number 2340 in Lecture Notes in Computer Science, pp. 279–289, Springer-Verlag, 2002.

[8] A. Ehrenfeucht, D. Prescott, G. Rozenberg, Computational aspects of gene (un)scrambling in ciliates, in: L. Landweber, E. Winfree (Eds.), *Evolution as Computation*, Springer-Verlag, Berlin, Heidelberg, 2001, pp. 45–86.

[9] A. Ehrenfeucht, D. Prescott, G. Rozenberg, Molecular operations for DNA processing in hypotrichous ciliates. *European Journal of Protistology*, 37 (2001) 241–260.

[10] S. Ginsburg, *Algebraic and Automata-Theoretic Properties of Formal Languages*, North-Holland Publishing Company, Amsterdam, 1975.

[11] S. Ginsburg, E.H. Spanier, Finite turn pushdown automata, *SIAM Journal Control* 4 (3) (1966) 429–453.

[12] E.M. Gurari, O. Ibarra, The complexity of decision problems for finite-turn multicounter machines, *Journal of Computer and System Sciences* 22 (1981) 220–229.

[13] T. Harju, O. Ibarra, J. Karhumäki, A. Salomaa, Some decision problems concerning semilinearity and commutation, *Journal of Computer and System Sciences* 65 (2002) 278–294.

[14] M. Harrison, *Introduction to Formal Language Theory*, Addison-Wesley, Reading, Ma, 1978.

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

[16] O. Ibarra, Reversal-bounded multicounter machines and their decision problems, *Journal of the ACM* 25 (1) (1978) 116–133.

[17] L. Kari, L. Landweber, Computational power of gene rearrangement, in: E. Winfree, D. Gifford (Eds.), *DNA5, DIMACS series in Discrete Mathematics and Theoretical Computer Science*, Vol. 54, American Mathematical Society, 2000, pp. 207–216.

[18] L. Landweber, L. Kari, The evolution of cellular computing: Nature's solution to a computational problem, in: L. Kari, H. Rubin, D. Wood (Eds.), *DNA4, BioSystems*, Vol. 52, Elsevier, 1999, pp. 3–13.

[19] D. Prescott, The unusual organization and processing of genomic DNA in hypotrichous ciliates, *Trends in Genet.* 8 (1992) 439–445.

[20] D. Prescott, Genome gymnastics: Unique modes of DNA evolution and processing in ciliates, *Nature Reviews Genetics* 1 (2000) 191–198.

[21] D. Prescott, A. Ehrenfeucht, G. Rozenberg, Template-guided recombination for IES elimination and unscrambling of genes in stichotrichous ciliates, *Journal of Theoretical Biology* 222 (2003) 323–330.