Useful Templates and Iterated Template-Guided DNA Recombination in Ciliates. *†

Mark Daley[1,2], Ian McQuillan[2]

[1] Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, S7N 5A9, Canada

[2] Department of Computer Science
University of Western Ontario
London, ON N6A 5B7, Canada
daley@csd.uwo.ca, imcquill@csd.uwo.ca

Abstract

The family of stichotrichous ciliates contains several single-celled organisms possessing a unique genetic mechanism: the ability to descramble genes which exist in a scrambled state in their genomes. We continue the theoretical investigation of the iterated template-guided recombination operation. This operation is suggested by the recombination of DNA strands based on template guides proposed by Prescott, Ehrenfeucht and Rozenberg. A variety of results is demonstrated including a study of computational power, characterizations and other abstract properties, such as a "pumping lemma". The notion of a

*This research was funded in part by institutional grants of the University of Saskatchewan and the Natural Sciences and Engineering Research Council of Canada.

†Published in Theory of Computing Systems (2006) 39: 619–633. https://doi.org/10.1007/s00224-005-1206-6

useful template is defined and forms a critical basis for much of the results demonstrated in the paper. The main result shows that every full AFL is closed under iterated template-guided recombination with regular templates.

1 Introduction

The stichotrichous ciliates are a family of single-celled organisms that contains some organisms which possess genomes with the curious property that certain genes exist in a scrambled form.

Unlike mammalian cells, which contain a single nucleus, every stichotrich has two separate nuclei: a functional macronucleus, which performs the "day-to-day" genetic chores of the cell, and an inert micronucleus which contains germline information. It is the micronucleus which contains scrambled genes.

Ciliates reproduce asexually, however they still do mate when nutrients become scarce. Specifically, when two ciliate cells conjugate, they destroy their macronuclei and exchange haploid micronuclear genomes. Each cell then builds a new functional macronucleus from the genetic material stored in the micronucleus. We have noted, however, that some of the genes stored in the micronucleus are stored in a scrambled order. Specifically, the micronuclear gene consists of fragments of the macronuclear gene in some permuted order. These fragments are called *macronuclear destined segments* or MDSs. Two MDSs which are consecutive in a macronuclear gene are both flanked by an identical short repeat segment in the micronuclear version of the gene. These repeat segments are called *pointers*. The cell must descramble-scramble the MDSs in order to create a functional gene which is capable of generating a protein. For more information on the biological process of gene descrambling-scrambling, we refer to [3, 10, 11, 12].

Several hypotheses as to how this descrambling-scrambling process takes place have been proposed in the literature. There are three primary theoretical models which have been investigated: the Kari-Landweber model [8, 9] which consists of binary inter- and intra-molecular recombination operations; the Ehrenfeucht, Harju, Petre, Prescott and Rozenberg model [5, 4, 6, 3] which consists of three unary operations inspired by intramolecular DNA recombination; and, most recently, a new model proposed by Prescott, Ehrenfeucht and Rozenberg [13, 2] based on the recombination of DNA strands guided by templates. We will formalize this latter model in the context of

formal language theory and study it here.

The basic action of the template-guided recombination operation is to take two DNA strands and splice them together using a third template strand. That is, for two strands of the form $u\alpha\beta d$ and $e\beta\gamma v$ (where $u, v, \alpha, \beta, \gamma, d, e$ are subsequences of a DNA strand) to be spliced together, we require a template of the form $\bar{\alpha}\bar{\beta}_1\bar{\beta}_2\bar{\gamma}$ where $\bar{\alpha}$ denotes a DNA sequence which is complementary to α, and $\beta = \beta_1\beta_2$. Specifically, the $\bar{\alpha}\bar{\beta}_1$ in the template will bind to the $\alpha\beta_1$ in the first strand and $\bar{\beta}_2\bar{\gamma}$ will bind to the $\beta_2\gamma$ in the second strand. The molecules then recombine with d and e being cleaved and removed, a new copy of the template $\bar{\alpha}\bar{\beta}\bar{\gamma}$ and the product of our recombination which is $u\alpha\beta\gamma v$.

In this paper we study a formal language theoretic model of the iterated version of this operation: iterated template-guided recombination. This is likely more biochemically realistic than the notion of an operation which is applied exactly once. In [2], we considered a comparison of that operation with iterated splicing schemes. In particular, we examined the capabilities of one to simulate the other. It was determined that iterated splicing could "almost always" simulate iterated template-guided recombination under weak conditions, but the reverse simulation could only occur if iterated splicing did not increase the capacity of the initial language family. In this way, iterated splicing is "almost always" more powerful.

Here, we introduce the notion of *useful templates*. Informally, a template word is useful on an initial language if it can be used as a template to generate any word, not necessarily new, when iterating the operation of template-guided recombination. A template language is useful on an initial language if every word is useful on the initial language. We show that every full AFL is closed under iterated template-guided recombination with useful template languages from the same full AFL (Theorem 4.1). In addition, we provide several characterizations of the iterated template-guided recombination operation when the template language is useful (Lemma 4.3, proof of Theorem 4.1).

Perhaps most interesting is a pumping lemma that we demonstrate for languages which are generated by iterated template-guided recombination (Lemma 4.2). It exposes some surprising necessary periodicity that generated languages must possess. Also, in our formalization, the operation of (iterated) template-guided recombination is parameterized by the minimum length of an MDS and a pointer (recall the meanings of MDSs and pointers above). In fact, the pumping lemma shows that if n is the sum of the

minimum lengths of an MDS and a pointer, then a template can be inserted between any two segments, one with the first n symbols of the template as subsequence and the other with the last n symbols of the template as subsequence. This shows that if the actual minimum length is too small, then there would be a large number of incorrect products. This is consistent with experimental evidence gathered to date, where the smallest MDS has been nine nucleotides long. We will clarify this in Section 4.

Further, we show using a proof without an effective construction that the subset of a regular template language which consists solely of useful words on any initial language (not even necessarily recursively enumerable) is also regular (Theorem 4.2). This can be combined with Theorem 4.1 to show that every full AFL is closed under iterated template-guided recombination with regular languages (Theorem 4.3).

Consequently, these results give us insight into the nature of this operation as both a biological process and a potential mechanism for *in vivo* computing.

2 Preliminaries

We refer to [14] for language theory preliminaries. Let Σ be a finite alphabet. We denote by Σ^* and Σ^+ the sets of all words and non-empty words, respectively, over Σ and the empty word by λ. For $k \in \mathbb{N}$, let $\Sigma^{\geq k}$ be all words of length at least k. A language L is any subset of Σ^*. Let $L^0 = \{\lambda\}$, $L^i = L^{i-1}L$, for all $i > 0$ and L^*, L^+ be catenation closure including and not including, respectively, the empty word as per the usual definitions.

Let $L, R \subseteq \Sigma^*$. We denote by $R^{-1}L = \{z \in \Sigma^* \mid yz \in L \text{ for some } y \in R\}$ and $LR^{-1} = \{z \in \Sigma^* \mid zy \in L \text{ for some } y \in R\}$. Let $x, y \in \Sigma^*$ and let $n \in \mathbb{N}$. We denote by $|x|$ the length of x. Let $x = a_1 \cdots a_m$, $a_i \in \Sigma$. We define $\Gamma^{\mathrm{p}}_n(x) = a_1 \cdots a_n$ if $m \geq n$ and λ otherwise and also $\Gamma^{\mathrm{s}}_n(x) = a_{m-n+1} \cdots a_m$ if $m \geq n$ and λ otherwise. We say x is a prefix of y, written $x \leq_{\mathrm{p}} y$, if there exists a word $w \in \Sigma^*$ such that $y = xw$. We say that x is a suffix of y, written $x \leq_{\mathrm{s}} y$, if there exists a word $w \in \Sigma^*$ such that $y = wx$. We say that x is an infix (or subword) of y, written $x \leq_{\mathrm{i}} y$, if there exist words $u, w \in \Sigma^*$ such that $y = uxw$. Furthermore, we let $\mathrm{pref}(L) = \{x \in \Sigma^* \mid x \leq_{\mathrm{p}} y, \text{ for some } y \in L\}$, $\mathrm{pref}_n(L) = \mathrm{pref}(L) \cap \Sigma^{\geq n}$, $\mathrm{suf}(L) = \{x \in \Sigma^* \mid x \leq_{\mathrm{s}} y, \text{ for some } y \in L\}$, $\mathrm{suf}_n(L) = \mathrm{suf}(L) \cap \Sigma^{\geq n}$, $\mathrm{inf}(L) = \{x \in \Sigma^* \mid x \leq_{\mathrm{i}} y, \text{ for some } y \in L\}$ and $\mathrm{inf}_n(L) = \mathrm{inf}(L) \cap \Sigma^{\geq n}$.

A *full AFL* is a language family (where a language family is defined as in

[1]) closed under homomorphism, inverse homomorphism, intersection with regular languages, union, concatenation and $*$. It is known that every full AFL is closed under nondeterministic gsm mappings, prefix, suffix, infix and left and right quotient with regular languages. We refer to [1, 7] for the theory of AFLs.

3 Template-guided recombination

We begin by defining the abstract formal version of the template-guided recombination operation described in [2].

Definition 3.1 *A* template-guided recombination system *(or TGR system) is a four-tuple $\varrho = (T, \Sigma, n_1, n_2)$ where Σ is a finite alphabet, $T \subseteq \Sigma^*$ is the template language, $n_1 \in \mathbb{N}$ is the minimum MDS length and $n_2 \in \mathbb{N}$ is the minimum pointer length.*

For a TGR system $\varrho = (T, \Sigma, n_1, n_2)$ and a language $L \subseteq \Sigma^*$, we define $\varrho(L) = \{w \in \Sigma^* \mid (x, y) \vdash_t^\varrho w$ for some $x, y \in L, t \in T\}$ where $(x, y) \vdash_t^\varrho w$ if and only if $x = u\alpha\beta d, y = e\beta\gamma v, t = \alpha\beta\gamma$ and $w = u\alpha\beta\gamma v$ for some $u, v, d, e \in \Sigma^*, \alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{\geq n_2}$. When there is no confusion, we denote $\vdash_t^\varrho$ by $\vdash_t$. We say that L is the base or initial language.

For language families $\mathcal{L}_1, \mathcal{L}_2$, we write $\pitchfork(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) = \{\varrho(L) \mid L \in \mathcal{L}_1, \varrho = (T, \Sigma, n_1, n_2)$ is a TGR system with $T \in \mathcal{L}_2\}$ and $\pitchfork(\mathcal{L}_1, \mathcal{L}_2) = \{\pitchfork(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) \mid n_1, n_2 \in \mathbb{N}\}$.

In [13], a constant C is defined such that $|\alpha|, |\gamma| > C$ in order to ensure the formation of sufficiently strong chemical bonds. The minimum MDS length above is general enough to cover any such constant. Likewise, [13] also defines constants D and E such that $D < |\beta| < E$. The minimum pointer length above is general enough to cover any such D. In addition, the constant E is shown to be irrelevant in the next, obvious, proposition from [2]. It states that we can always assume that the β subword of a template is of length n_2.

Proposition 3.1 *Let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system and let $x, y \in \Sigma^*$ and $t \in T$. Then $(x, y) \vdash_t w$ if and only if $x = u\alpha\beta d, y = e\beta\gamma v, t = \alpha\beta\gamma$ and $w = u\alpha\beta\gamma v$ for some $u, v, d, e \in \Sigma^*, \alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{n_2}$.*

In the sequel, we shall thus assume, without loss of generality, that β is of length n_2.

Next, we define iterated template-guided recombination:

Let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system and let $L \subseteq \Sigma^*$. Then we generalize ϱ to an iterated operation $\varrho^*(L)$ as follows:

$$\begin{aligned} \varrho^0(L) = &\quad L, \\ \varrho^{i+1}(L) = &\quad \varrho^i(L) \cup \varrho(\varrho^i(L)), i \geq 0 \\ \varrho^*(L) = &\quad \bigcup_{i=0}^{\infty} \varrho^i(L). \end{aligned}$$

Further, for language families $\mathcal{L}_1, \mathcal{L}_2$, define $\pitchfork^*(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) = \{\varrho^*(L) \mid L \in \mathcal{L}_1, \varrho = (T, \Sigma, n_1, n_2)$ is a TGR system with $T \in \mathcal{L}_2\}$ and also let $\pitchfork^*(\mathcal{L}_1, \mathcal{L}_2) = \{\pitchfork^*(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) \mid n_1, n_2 \in \mathbb{N}\}$.

It is also clear, and known from [2], that applying iterated template-guided recombination to a language family yields a family which contains the original language family.

Lemma 3.1 *Let $\mathcal{L}_1, \mathcal{L}_2$ be language families and let $n_1, n_2 \in \mathbb{N}$. Then $\mathcal{L}_1 \subseteq \pitchfork^*(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2)$.*

4 Useful templates

Next, we define the concepts of a useful template and a useful system which we will use throughout the rest of this paper.

Definition 4.1 *Let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system and let $L \subseteq \Sigma^*$. A word $t \in T$ is* useful *on (L, ϱ) (or simply useful if the context is understood), if there exist words $x = u\alpha\beta d, y = e\beta\gamma v \in \varrho^*(L)$ such that $t = \alpha\beta\gamma, \alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{n_2}, u, d, e, v \in \Sigma^*$. If every word in T is useful on (L, ϱ), then ϱ is useful on L.*

Intuitively, a template word is useful if it can be used as a template to produce any word when applying the template-guided recombination operation to a language.

The following lemma will be used many times in the sequel.

Lemma 4.1 *Let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system. A word $t \in T$ is useful on (L, ϱ) if and only if $|t| \geq 2n_1 + n_2$ and there exists a word $w \in \varrho^*(L)$ such that $t \leq_{\mathrm{i}} w$.*

Proof. Assume that t is useful on (L, ϱ). Then, there exist words $x = u\alpha\beta d, y = e\beta\gamma v \in \varrho^*(L), \alpha\beta\gamma = t, \alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{n_2}, u, v, d, e \in \Sigma^*$. Thus, $w = u\alpha\beta\gamma v \in \varrho^*(L), |\alpha\beta\gamma| \geq 2n_1 + n_2$ and $t \leq_{\mathrm{i}} w$.

Assume that $|t| \geq 2n_1 + n_2$ and that there exists a word $w \in \varrho^*(L)$ such that $t \leq_{\mathrm{i}} w$. Thus, $w = utv, t = \alpha\beta\gamma$, where $\alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{n_2}, u, v \in \Sigma^*$. If we let $d = \gamma v$ and $e = u\alpha$, then $w = x = u\alpha\beta d, w = y = e\beta\gamma v$, and $x, y \in \varrho^*(L)$. Hence, t is useful on (L, ϱ). ■

Next, we describe a type of "pumping lemma" when the TGR system is useful.

Let $L \subseteq \Sigma^*$ and let $n \in \mathbb{N}$. Then, let

$$P_{a_1,\ldots,a_n}(L) = \mathrm{pref}(L)(a_1 \cdots a_n)^{-1}$$

and

$$S_{a_1,\ldots,a_n}(L) = (a_1 \cdots a_n)^{-1}\mathrm{suf}(L),$$

for each $a_1, \ldots, a_n \in \Sigma$. So, $P_{a_1,\ldots,a_n}(L)$ consists of all prefixes of words of L ending in $a_1 \cdots a_n$ with the final $a_1 \cdots a_n$ removed and $S_{a_1,\ldots,a_n}(L)$ consists of all suffixes of words of L starting with $a_1 \cdots a_n$ with the starting $a_1 \cdots a_n$ removed where a_1 through a_n are single letters of the alphabet. It is clear that every full AFL $\mathcal{L}$ is closed under $P_{a_1,\ldots,a_n}$ and $S_{a_1,\ldots,a_n}$, for every $a_1, \ldots, a_n \in \Sigma$ as one can construct a nondeterministic gsm simulating both.

Lemma 4.2 *Let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system and let $n = n_1 + n_2$. Let $L \subseteq \Sigma^*$ with $t \in T$ useful on (L, ϱ) and let $L' = \varrho^*(L)$. Then*

$$P_{a_1,\ldots,a_n}(L')tS_{b_1,\ldots b_n}(L') \subseteq L'$$

where $|t| \geq 2n_1 + n_2, \Gamma^{\mathrm{p}}_n(t) = a_1 \cdots a_n, \Gamma^{\mathrm{s}}_n(t) = b_1 \cdots b_n$.

Proof. Let $u \in P_{a_1,\ldots,a_n}(L')$ and let $v \in S_{b_1,\ldots,b_n}(L')$. So, $ua_1 \cdots a_n d \in L'$ and $eb_1 \cdots b_n v \in L'$, for some $d, e \in \Sigma^*$. We know $t \in T$ is useful on (L, ϱ), so by Lemma 4.1, there exists a word $w \in L', t \leq_{\mathrm{i}} w$ and $|t| \geq 2n_1 + n_2$. We can rewrite $w = ya_1 \cdots a_n x_1 z = yx_2 b_1 \cdots b_n z$, for some $x_1, x_2, y, z \in \Sigma^*$ where $x_2 b_1 \cdots b_n = a_1 \cdots a_n x_1 = t$. But,

$$(ua_1 \cdots a_n d, ya_1 \cdots a_n x_1 z) \vdash_t ua_1 \cdots a_n x_1 z \in L'.$$

Furthermore,

$$((ua_1 \cdots a_n x_1 z = ux_2 b_1 \cdots b_n z), eb_1 \cdots b_n v) \vdash_t ux_2 b_1 \cdots b_n v = utv \in L'.$$

Hence, $utv \in L'$ and the claim follows. ■

Essentially, what this means is that if t starts with $a_1 \cdots a_n$ and ends with $b_1 \cdots b_n$, then t can be inserted at any spot between any prefix of a word in L' ending in $a_1 \cdots a_n$ and any suffix of a word in L' starting with $b_1 \cdots b_n$. Thus, in some sense, as long as we start with the knowledge that a template word is useful, only n symbols of each side guide insertion. This will become evident with the following example.

Example 4.1 *Let $\Sigma = \{a, b, c_1, c_2, c_3, c_4, c_5, c_6\}$ be an alphabet. Also, consider $L = \{c_1aac_2c_3, c_4aac_5c_3, c_3bbc_6, abbc_1\}$ and $T = \{aac_2c_3bb\}$ where $\varrho = (T, \Sigma, 1, 1)$ is a TGR system. Then*

$$(c_1aac_2c_3, c_3bbc_6) \vdash_{aac_2c_3bb} c_1aac_2c_3bbc_6$$

and so aac_2c_3bb is useful and so is ϱ. But then,

$$(c_4aac_5c_3, c_1aac_2c_3bbc_6) \vdash_{aac_2c_3bb} c_4aac_2c_3bbc_6.$$

Indeed,

$$\varrho^*(L) = L \cup \{c_1aac_2c_3bbc_6, c_4aac_2c_3bbc_6, c_1aac_2c_3bbc_1, c_4aac_2c_3bbc_1\},$$

and observe that $P_{a,a}(\varrho^*(L)) = \{c_1, c_4\}$, $S_{b,b}(\varrho^*(L)) = \{c_1, c_6\}$.

The "pumping lemma" above shows that when we iterate the template-guided recombination operation, only n symbols are necessary on either side of the template to guide insertion of the template. Thus, assuming template-guided recombination is the correct hypothesis, if the actual value of n is too small, then there would be a large number of incorrect products of iterated template-guided recombination. To date, the smallest MDS found experimentally has been nine nucleotides long.

Not only can this be used as a type of "pumping lemma", we can also use it to characterize the languages $\varrho^*(L), L \subseteq \Sigma^*$, where $\varrho = (T, \Sigma, n_1, n_2)$ is a TGR system with $n = n_1 + n_2$, whenever ϱ is useful on L.

Let

$$\begin{aligned}
L_\varrho^{(0)} &= L \\
L_\varrho^{(i+1)} &= L_\varrho^{(i)} \cup \{utv \mid t \in T, \Gamma_n^{\mathrm{p}}(t) = a_1 \cdots a_n, \Gamma_n^{\mathrm{s}}(t) = b_1 \cdots b_n, \\
&\quad u \in P_{a_1,\ldots,a_n}(L_\varrho^{(i)}), v \in S_{b_1,\ldots,b_n}(L_\varrho^{(i)})\} \text{ for all } i \geq 0, \\
L_\varrho^{(*)} &= \bigcup_{i=0}^{\infty} L_\varrho^{(i)}.
\end{aligned}$$

Lemma 4.3 *Let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system and let $L \subseteq \Sigma^*$ with ϱ useful on L. Then*

$$\varrho^*(L) = L_\varrho^{(*)}.$$

Proof. Let $n = n_1 + n_2$.

"$\subseteq$" To show $\varrho^*(L) \subseteq L_\varrho^{(*)}$ it suffices to prove, by induction on i, that $\varrho^i(L) \subseteq L_\varrho^{(i)}$ for every $i \geq 0$. This is obvious for $i = 0$ (since both sets equal L). Now assume that it holds for $m \geq 0$, i.e., $\varrho^m(L) \subseteq L_\varrho^{(m)}$, and let $w \in \varrho^{m+1}(L) - \varrho^m(L)$. Thus, $(x, y) \vdash_t w$ for some $x, y \in \varrho^m(L) \subseteq L_\varrho^{(m)}, t \in T$, by the inductive hypothesis. Thus, $w = u\alpha\beta\gamma v, x = u\alpha\beta d, y = e\beta\gamma v, \alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{n_2}, u, v, d, e \in \Sigma^*, \alpha\beta\gamma = t$. So, if $a_1 \cdots a_n$ are the first n letters of $\alpha\beta$ and $b_1 \cdots b_n$ are the last n letters of $\beta\gamma$, then $u \in P_{a_1,\ldots,a_n}(L_\varrho^{(m)})$ and $v \in S_{b_1,\ldots,b_n}(L_\varrho^{(m)})$. Hence, $w = utv \in L_\varrho^{(m+1)}$. This proves that $\varrho^{m+1}(L) \subseteq L_\varrho^{(m+1)}$.

"$\supseteq$" To show $L_\varrho^{(*)} \subseteq \varrho^*(L)$, we prove by induction on i that $L_\varrho^{(i)} \subseteq \varrho^*(L)$. This is, again, obvious for $i = 0$. Assuming that it holds for $m \geq 0$, let $w \in L_\varrho^{(m+1)} - L_\varrho^{(m)}$. Thus, $w = utv$, where $t \in T$ with $\Gamma_n^{\mathrm{p}}(t) = a_1 \cdots a_n, \Gamma_n^{\mathrm{s}}(t) = b_1 \cdots b_n$ and $u \in P_{a_1,\ldots,a_n}(L_\varrho^{(m)}) \subseteq P_{a_1,\ldots,a_n}(\varrho^*(L)), v \in S_{b_1,\ldots,b_n}(L_\varrho^{(m)}) \subseteq S_{b_1,\ldots,b_n}(\varrho^*(L))$. By the inductive hypothesis and by Lemma 4.2, we see that $utv \in \varrho^*(L)$. This proves that $L_\varrho^{(m+1)} \subseteq \varrho^*(L)$. ∎

We use this to prove one of the main results of the paper. Indeed, we show that every full AFL is closed under iterated template-guided recombination with useful templates from the same full AFL. The construction provides an interesting characterization as well.

Theorem 4.1 *Let $\mathcal{L}$ be a full AFL, $\varrho = (T, \Sigma, n_1, n_2)$ a TGR system and let $L, T \in \mathcal{L}, L \subseteq \Sigma^*$, and assume that ϱ is useful on L. Then $\varrho^*(L) \in \mathcal{L}$.*

Proof. Let $n = n_1 + n_2$ and $V_\$ = \{\$_{L,p}, \$_{L,i}, \$_{L,s}, \$_T, \$_{T,s}, \$_{T,p}, \$_{T,i}\}$, all new symbols. Also, for each $x \in \Sigma^*$,we define $\acute{x} = \{\Sigma^n\}^{-1}\{x\}$ and $\grave{x} = \{x\}\{\Sigma^n\}^{-1}$.

Let $L_1 = (\$_{L,p}\mathrm{pref}_n(L) \cup \$_{L,i}\mathrm{inf}_n(L) \cup \$_{L,s}\mathrm{suf}_n(L) \cup \$_T T \cup \$_{T,s}\mathrm{suf}_n(T) \cup \$_{T,p}\mathrm{pref}_n(T) \cup \$_{T,i}\mathrm{inf}_n(T))^+$.

It is clear that $L_1 \in \mathcal{L}$ since every full AFL is closed under concatenation with a new symbol, union, prefix, suffix, infix, + and intersection with regular languages.

Next, we intersect L_1 with a regular language, R, which enforces that the symbol from $V_\$$ that appears

first must be $\$_{L,p}$,
directly after $\$_{L,p}$ must be $\$_{T,p}$ or $\$_T$,
directly after $\$_{L,i}$ must be $\$_{T,p}$ or $\$_T$,
directly after $\$_{T,p}$ must be $\$_{T,p}$ or $\$_T$,
directly after $\$_{T,i}$ must be $\$_{T,p}$ or $\$_T$,
directly after $\$_T$ must be $\$_{L,s}, \$_{T,s}, \$_{L,i}$ or $\$_{T,i}$,
directly after $\$_{T,s}$ must be $\$_{L,s}, \$_{T,s}, \$_{L,i}$ or $\$_{T,i}$,
last must be $\$_{L,s}$.

Let $L_2 = L_1 \cap R$ be this new language. Again, $L_2 \in \mathcal{L}$. It is easy to see that, equivalently, R enforces that the symbol from $V_\$$ that appears

first must be $\$_{L,p}$,
directly before $\$_{L,s}$ must be $\$_{T,s}$ or $\$_T$,
directly before $\$_{L,i}$ must be $\$_{T,s}$ or $\$_T$,
directly before $\$_T$ must be $\$_{L,p}, \$_{T,p}, \$_{L,i}$ or $\$_{T,i}$,
directly before $\$_{T,p}$ must be $\$_{L,p}, \$_{T,p}, \$_{L,i}$ or $\$_{T,i}$,
directly before $\$_{T,s}$ must be $\$_{T,s}$ or $\$_T$,
directly before $\$_{T,i}$ must be $\$_{T,s}$ or $\$_T$,
last must be $\$_{L,s}$.

Next, we obtain L_3 by intersecting L_2 with a regular language that enforces that the last n symbols from Σ before each marker are the same as the first n after the marker. That is, $L_3 = \{\$_1 y_1 \$_2 y_2 \cdots \$_m y_m \in L_2 \mid \$_i \in V_\$, y_i \in \Sigma^+$, and $\Gamma^{\mathrm{s}}_n(y_i) = \Gamma^{\mathrm{p}}_n(y_{i+1})$ for every i, $1 \leq i < m\}$.

Finally, let M be a deterministic gsm-mapping which erases all symbols from $V_\$$, and erases the first n letters from Σ after each symbol from $V_\$$, except the first. That is, if $\alpha = \$_1 y_1 \$_2 y_2 \cdots \$_m y_m$ then $M(\alpha) = y_1 y_2' \cdots y_m'$. Let $L' = M(L_3) \cup L$. Clearly, $L' \in \mathcal{L}$ as every full AFL is closed under union and gsm mappings. Indeed, we will show that $L' = \varrho^*(L)$.

Claim 4.1 $L' \subseteq \varrho^*(L)$.

Proof. Let $w \in L'$. If $w \in L$, then $w \in \varrho^*(L)$ and we are done. Assume then, that $w \in L' - L$. Let α be a string in L_3 such that $M(\alpha) = w$. Let $\alpha = \$_1 y_1 \$_2 y_2 \cdots \$_q y_q$ with $\$_i \in V_\$$ and $y_i \in \Sigma^+$ for $1 \leq i \leq q$. Observe that

$q \geq 3$ since $\$_1 = \$_{L,p}, \$_q = \$_{L,s}$ and $\$_2$ can be neither. Furthermore, for each $i, 1 \leq i \leq q$, let $w_i = M(\$_1 y_1 \cdots \$_i y_i)$. In particular, $w_q = w$. Also, for each i, let $a_{i,1}, \ldots, a_{i,n}, b_{i,1}, \ldots, b_{i,n} \in \Sigma$ satisfy $\Gamma^{\mathrm{p}}_n(y_i) = a_{i,1} \cdots a_{i,n}$ and $\Gamma^{\mathrm{s}}_n(y_i) = b_{i,1} \cdots b_{i,n}$. Notice that, by the construction,

$$b_{i,1} \cdots b_{i,n} = a_{i+1,1} \cdots a_{i+1,n}, \tag{1}$$

for every $i, 1 \leq i \leq q-1$.

We will show by induction on i, $1 \leq i \leq q$, that $w_i \leq_{\mathrm{p}} v_i$ for some $v_i \in \varrho^*(L)$ and that $w_q = w \in \varrho^*(L)$.

This is true for $i = 1$ as the first segment y_1 is a prefix of some word in $L \subseteq \varrho^*(L)$.

Let m be an integer such that $1 \leq m < q$. Assume, by way of induction, that $w_m \leq_{\mathrm{p}} v_m$ for some $v_m \in \varrho^*(L)$. We let $u_m \in \Sigma^*$ be such that $v_m = w_m u_m$. Also, we let $a_1 = a_{m+1,1} = b_{m,1}, \ldots, a_n = a_{m+1,n} = b_{m,n}$.

case 1: Assume that either $\$_{m+1} = \$_{T,p}$ (and $y_{m+1} \in \mathrm{pref}(T)$) or $\$_{m+1} = \$_T$ (and $y_{m+1} \in T$). In either case, there exist $x_1 \in T$ such that $y_{m+1} \leq_{\mathrm{p}} x_1$ and another word $x_2 \in \varrho^*(L)$, such that $x_1 \leq_{\mathrm{i}} x_2$, by Lemma 4.1. We rewrite $v_m = w_m u_m = \grave{w_m} b_{m,1} \cdots b_{m,n} u_m = \grave{w_m} a_1 \cdots a_n u_m$ and also $x_2 = e x_1 v = e a_1 \cdots a_n \acute{x_1} v, e, v \in \Sigma^*$. Thus,

$$((v_m = \grave{w_m} a_1 \cdots a_n u_m), (x_2 = e a_1 \cdots a_n \acute{x_1} v)) \vdash_{(x_1 = a_1 \cdots a_n \acute{x_1})} (w_m \acute{x_1} v = \grave{w_m} x_1 v)$$

and so there exists a word, $w_m \acute{x_1} v \in \varrho^*(L)$ such that $w_{m+1} = w_m \acute{y}_{m+1} \leq_{\mathrm{p}} w_m \acute{x_1} v$.

case 2: Assume that $\$_{m+1}$ is equal to $\$_{L,s}, \$_{L,i}, \$_{T,s}$ or $\$_{T,i}$.

case 2a: Assume that $\$_{m+1} = \$_{L,s}$ and thus $y_{m+1} \in \mathrm{suf}(L)$. Let $e y_{m+1} \in L, e \in \Sigma^*$. By the construction, either $\$_m$ is equal to $\$_{T,s}$ or $\$_T$. Furthermore, since $\$_1 = \$_{L,p}$, there must exist an integer j, such that $\$_j = \$_T, 2 \leq j \leq m$ and for every $k, j < k \leq m, \$_k = \$_{T,s}$. For each word $y_k, j \leq k \leq m$, let $t_k = r_k y_k \in T$, for some $r_k \in \Sigma^*$ and let $r_j = \lambda$. For each $t_k, j \leq k \leq m$, let $s_k = \mu_k t_k \nu_k \in \varrho^*(L)$, for some $\mu_k, \nu_k \in \Sigma^*$, which must exist by Lemma 4.1. Then $s_k = \mu_k r_k y_k \nu_k$. Thus,

$$\begin{aligned}
&(\mu_m r_m y_m \nu_m, (e y_{m+1} = e a_1 \cdots a_n \acute{y}_{m+1})) \vdash_{(r_m y_m = r_m \grave{y_m} a_1 \cdots a_n)} \mu_m r_m y_m \acute{y}_{m+1} \\
&(\mu_{m-1} r_{m-1} y_{m-1} \nu_{m-1}, \mu_m r_m y_m \acute{y}_{m+1}) \vdash_{r_{m-1} y_{m-1}} \mu_{m-1} r_{m-1} y_{m-1} \acute{y}_m \acute{y}_{m+1} \\
&\quad \vdots \\
&(\mu_j r_j y_j \nu_j, \mu_{j+1} r_{j+1} y_{j+1} \acute{y}_{j+2} \acute{y}_{j+3} \cdots \acute{y}_{m+1}) \vdash_{r_j y_j} \mu_j r_j y_j \acute{y}_{j+1} \cdots \acute{y}_{m+1}
\end{aligned} \tag{2}$$

since the last n letters of y_k must always be equal to the first n of y_{k+1}. Then by Lemma 4.2, $P_{a_{j,1},\ldots,a_{j,n}}(\varrho^*(L))y_j S_{b_{j,1},\ldots,b_{j,n}}(\varrho^*(L)) \subseteq \varrho^*(L)$. Notice that $w_{j-1} \leq_{\rm p} w_m \leq_{\rm p} v_m \in \varrho^*(L)$, by the inductive hypothesis and consequently $\grave{w}_{j-1} \in P_{a_{j,1},\ldots,a_{j,n}}(\varrho^*(L))$. In addition, $\mu_j r_j y_j \acute{y}_{j+1} \cdots \acute{y}_{m+1} \in \varrho^*(L)$ and so $\acute{y}_{j+1} \cdots \acute{y}_{m+1} \in S_{b_{j,1},\ldots,b_{j,n}}(\varrho^*(L))$. Hence,

$$\grave{w}_{j-1} y_j \acute{y}_{j+1} \cdots \acute{y}_{m+1} = w_{j-1} \acute{y}_j \acute{y}_{j+1} \cdots \acute{y}_{m+1} = w_{m+1} \in \varrho^*(L).$$

cases 2b,c,d: Case b ($\$_{m+1} = \$_{L,i}$) is similar to case a except replace ey_{m+1} in (2) by $ey_{m+1}e'$ where $e, e' \in \Sigma^*$. We now obtain the same recombination results followed by e', and $w_{m+1}e' \in \varrho^*(L)$. Thus, w_{m+1} is a prefix of some word in $\varrho^*(L)$. Cases c ($\$_{m+1} = \$_{T,s}$) and d ($\$_{m+1} = \$_{T,i}$) are also similar, except we replace ey_{m+1} in (2) with $s_{m+1} = \mu_{m+1} t_{m+1} \nu_{m+1} \in \varrho^*(L)$ where $t_{m+1} = r_{m+1} y_{m+1} f_{m+1} \in T, r_{m+1}, f_{m+1} \in \Sigma^*$ which must exist by Lemma 4.1. We then obtain the same recombination results followed by $f_{m+1}v_{m+1}$, and $w_{m+1}f_{m+1}v_{m+1} \in \varrho^*(L)$.

Thus, by way of induction, it follows that $w_q = w \in \varrho^*(L)$. ■

Claim 4.2 $\varrho^*(L) \subseteq L'$.

Proof. By Lemma 4.3 it suffices to show, by induction on i, that $L_\varrho^{(i)} \subseteq L'$ for every $i \geq 0$. For $i = 0$ this is obvious because $L_\varrho^{(0)} = L$. Assume now, for $m \geq 0$, that $L_\varrho^{(m)} \subseteq L'$ and let $w \in L_\varrho^{(m+1)} - L_\varrho^{(m)}$. Thus, we can write $w = x_1 t x_2, t \in T, \Gamma_n^{\rm p}(t) = a_1 \cdots a_n, \Gamma_n^{\rm s}(t) = b_1 \cdots b_n$, for some $a_1, \ldots, a_n, b_1, \ldots, b_n \in \Sigma$, $x_1 \in P_{a_1,\ldots,a_n}(L_\varrho^{(m)})$ and $x_2 \in S_{b_1,\ldots,b_n}(L_\varrho^{(m)})$. Then, there exist $r_1 \in L_\varrho^{(m)}$ such that $r_1 = x_1 a_1 \cdots a_n s_1, s_1 \in \Sigma^*$, and $r_2 \in L_\varrho^{(m)}$ such that $r_2 = s_2 b_1 \cdots b_n x_2, s_2 \in \Sigma^*$. Next, let $\alpha_1 = \$_{L,p} x_1 a_1 \cdots a_n$ if $r_1 \in L$ and $\alpha_1 \in L_3$ such that $M(\alpha_1) = r_1$ otherwise, which must exist by the inductive hypothesis. Also, let $\alpha_2 = \$_{L,s} b_1 \cdots b_n x_2$ if $r_2 \in L$ and $\alpha_2 \in L_3$ such that $M(\alpha_2) = r_2$ otherwise, which also must exist by the inductive hypothesis. We rewrite $\alpha_1 = \$_{1,1} y_{1,1} \$_{1,2} y_{1,2} \cdots \$_{1,q_1} y_{1,q_1}$ and $\alpha_2 = \$_{2,1} y_{2,1} \$_{2,2} y_{2,2} \cdots \$_{2,q_2} y_{2,q_2}$, where $y_{1,j}, y_{2,k} \in \Sigma^+, \$_{1,j}, \$_{2,k} \in V_\$, 1 \leq j \leq q_1, 1 \leq k \leq q_2$. Let m_1 be the smallest integer such that $x_1 a_1 \cdots a_n \leq_{\rm p} M(\$_{1,1} y_{1,1} \cdots \$_{1,m_1} y_{1,m_1})$. Let m_2 be the largest integer such that $b_1 \cdots b_n x_2 \leq_{\rm s} M(\$_{2,m_2} y_{2,m_2} \cdots \$_{2,q_2} y_{2,q_2})$.

Consider the string

$$\gamma_1 = \$_{1,1} y_{1,1} \cdots \$_{1,m_1-1} y_{1,m_1-1} \$_{1,m_1} e_1,$$

where $M(\gamma_1) = x_1 a_1 \cdots a_n, e_1 \in \Sigma^*$, $e_1 \leq_{\mathrm{p}} y_{1,m_1}$. Furthermore, consider

$$\gamma_2 = \$_{2,m_2} e_2 \$_{2,m_2+1} y_{2,m_2+1} \cdots \$_{2,q_2} y_{2,q_2},$$

where $M(\gamma_2) = b_1 \cdots b_n x_2, e_2 \in \Sigma^*$, $e_2 \leq_{\mathrm{s}} y_{2,m_2}$. Notice that $|e_1| \geq n$ and $|e_2| \geq n$ since m_1 is the smallest such integer and m_2 is the largest. Finally, consider $\gamma_1 \$_T t \gamma_2$. It is clear that $M(\gamma_1 \$_T t \gamma_2) = w$, however we must ensure that $\gamma_1 \$_T t \gamma_2 \in L_3$.

If $\$_{1,m_1}$ is equal to either $\$_{L,p}, \$_{T,p}, \$_{L,i}$ or $\$_{T,i}$ and also $\$_{2,m_2}$ is equal to either $\$_{L,s}, \$_{T,s}, \$_{L,i}$ or $\$_{T,i}$, then $\gamma_1 \$_T t \gamma_2 \in L_3$ for the following reasons: $\$_T$ can appear directly after any of the first set of symbols (see the definition of R) and before any of the second set (see the equivalent definition of R), if y_{1,m_1} is a prefix (or infix, respectively) of some word in L, then so is e_1, if y_{1,m_1} is a prefix (or infix, respectively) of some word in T, then so is e_1, if y_{2,m_2} is a suffix (respectively infix) of some word in L, then so is e_2 and if y_{2,m_2} is a suffix (respectively, infix) of some word in T, then so is e_2. We will consider the remaining cases by making the following changes to the string $\gamma_1 \$_T t \gamma_2$. If $\$_{1,m_1} = \$_T$, then we change the symbol $\$_{1,m_1}$ to $\$_{T,p}$ as the symbols from $V_\$$ that can appear directly before $\$_{T,p}$ are the same as those of $\$_T$ and also e_1 is a prefix of y_{1,m_1}. For similar reasons, if $\$_{1,m_1}$ is $\$_{T,s}$ or $\$_{L,s}$, then we change it into $\$_{T,i}$ or $\$_{L,i}$, respectively. And similarly if $\$_{2,m_2}$ is $\$_T$, $\$_{T,p}$ or $\$_{L,p}$, then we change it into $\$_{T,s}$, $\$_{T,i}$ or $\$_{L,i}$, respectively.

After these changes to $\gamma_1 \$_T t \gamma_2$, we denote the resulting string by x. We see that $x \in L_3$ for the reasons mentioned above. Moreover, $M(\gamma_1 \$_T t \gamma_2) = M(x) = w \in L'$. Hence, by way of induction, it follows that $w \in L'$ and so $L_\varrho^{(m+1)} \subseteq L'$. ■

Hence, by Claims 4.1 and 4.2, it is immediate that $\varrho^*(L) \in \mathcal{L}$. ■

Let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system with $L \subseteq \Sigma^*$. Even though ϱ is not necessarily useful on L, it is obvious that there exists a subset T_u of T and a TGR system $\varrho_u = (T_u, \Sigma, n_1, n_2)$ which is useful on L. In fact, $T_u = \{t \in T \mid t \text{ is useful on } (L, \varrho)\}$. We call this subset *the useful subset* of ϱ on L (or just the useful subset if the context is understood) and we call ϱ_u the useful subsystem of ϱ on L. Thus, attention should now turn to finding the useful subset of the template language whenever possible.

We have been unable, as yet, to effectively determine the useful subset of a template language when it is a regular language. However, the next result shows that indeed, if L is any language at all (not even necessarily recursively

enumerable), and T is a regular language, then the useful subset of T on L is also regular.

Theorem 4.2 *Let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system and let $L \subseteq \Sigma^*$. Let T_u be the useful subset of ϱ on L. If T is a regular language, then T_u is also regular*[1].

Proof. Let $n = 2n_1 + n_2 - 1$ and $R = \{t \in T \mid |t| \geq 2n_1 + n_2\}$. Since $t \in T_u$ implies that $|t| \geq 2n_1 + n_2$, it follows that $T_u \subseteq R$. Let $M = (Q, \Sigma, q_0, F, \delta)$ be a deterministic finite automaton accepting R where $\delta : Q \times \Sigma \to Q$ is a partial function, extended to a partial function $\delta : Q \times \Sigma^* \to Q$ in the usual way. Let

$$f(q, a_1, \ldots, a_n) = \{v \mid v \in \inf(T_u), v \in a_1 \cdots a_n \Sigma^*, \delta(q, v) \in F\},$$

for each $q \in Q, a_1, \ldots, a_n \in \Sigma$ (these sets are not effectively constructed). This set consists of all infixes of some useful template which starts with $a_1 \cdots a_n$ and enters a final state starting in q using δ. We will create a new deterministic finite automaton $M' = (Q', \Sigma, q_0', F', \delta')$ by making the following modifications to M. In the finite control of M', it simulates M and also remembers the previous $n-1$ states entered and the last $n-1$ input symbols that were read. For any set A, we write $[A]^n$ to denote $A \times ... \times A$ (n times): the set of all sequences of elements of A of length n. Formally, $Q' = ([Q \cup \{\lambda\}]^n \times [\Sigma \cup \{\lambda\}]^{n-1}), F' = [Q]^{n-1} \times F \times [\Sigma]^{n-1}, q_0' = ([\lambda]^{n-1}, q_0, [\lambda]^{n-1})$ and δ' is defined as follows:

For every transition $\delta(q, a) = p$, with $p, q \in Q, a \in \Sigma$, we define both $\delta'((\lambda, p_1, \ldots, p_{n-1}, \lambda, b_1, \ldots, b_{n-2}), a) = (p_1, \ldots, p_{n-1}, p, b_1, \ldots, b_{n-2}, a)$ where $p_{n-1} = q, p_1, \ldots, p_{n-2} \in Q \cup \{\lambda\}, b_1, \ldots, b_{n-2} \in \Sigma \cup \{\lambda\}$ and also we define $\delta'((q_1, \ldots, q_{n-1}, q, a_1, \ldots, a_{n-1}), a) = (q_2, \ldots, q_{n-1}, q, p, a_2, \ldots, a_{n-1}, a)$, for each $q_1, \ldots, q_{n-1} \in Q, a_1, \ldots, a_{n-1} \in \Sigma$ iff $f(q_1, a_1, \ldots, a_{n-1}, a) \neq \emptyset$ (we have not given an effective procedure to decide this property). Let T' be the language accepted by M'. We claim that $T_u = T'$.

"$\subseteq$" Let $t = a_1 a_2 \cdots a_m \in T_u \subseteq R$ with $a_i \in \Sigma$ and let q_i, $0 \leq i \leq m$, satisfy $\delta(q_j, a_{j+1}) = q_{j+1}$, for all j, $0 \leq j < m$ with $q_m \in F$. We will show by induction that for every l, $1 \leq l \leq m$, $\delta'(q_0', a_1 a_2 \cdots a_l) = (q_{l-n+1}, \ldots, q_{l-1}, q_l, a_{l-n+2}, \ldots, a_l)$ where $q_i = a_j = \lambda$ for all $i < 0, j < 1$ (of which there are at most n of each). By the construction of M', $\delta'(q_0', a_1) =$

[1]This is not an effective construction.

$([\lambda]^{n-2}, q_0, q_1, [\lambda]^{n-2}, a_1)$. Let k be an integer such that $1 \leq k < m$. Assume, by induction, that $\delta'(q'_0, a_1a_2\cdots a_k) = (q_{k-n+1}, \ldots, q_{k-1}, q_k, a_{k-n+2}, \ldots, a_k)$. If $k+1 < n$, then $\delta'(q'_0, a_1a_2\cdots a_ka_{k+1}) = (q_{k-n+2}, \ldots, q_{k+1}, a_{k-n+3}, \ldots, a_{k+1})$ by the construction. Assume that $k+1 \geq n$. Since $a_1a_2\cdots a_m \in T_u$, we can conclude that $f(q_{k-n+1}, a_{k-n+2}, \ldots, a_{k+1}) \neq \emptyset$ as $a_{k-n+2}\cdots a_m \in f(q_{k-n+1}, a_{k-n+2}, \ldots, a_{k+1})$. Thus we obtain,

$$\delta'(q'_0, a_1a_2\cdots a_{k+1}) = (q_{k-n+2}, \ldots, q_{k+1}, a_{k-n+3}, \ldots, a_{k+1}).$$

Hence, by induction, $\delta'(q'_0, a_1a_2\cdots a_m) = (q_{m-n+1}, \ldots, q_m, a_{m-n+2}, \ldots, a_m)$, $q_m \in F, (q_{m-n+1}, \ldots, q_m, a_{m-n+2}, \ldots, a_m) \in F'$ and $t \in T'$.

"$\supseteq$" Let $t = a_1a_2\cdots a_m \in T'$, with $a_i \in \Sigma$. It follows that $t \in T$ and $m \geq n$ by the definition of R. By the definition of M', there must exist $q_0, q_1, \ldots, q_m \in Q$ with $q_m \in F$ and $q_{-n+2} = \cdots = q_{-1} = a_{-n+3} = \cdots = a_0 = \lambda$ such that $\delta(q_j, a_{j+1}) = q_{j+1}$ for all j, $0 \leq j < m$, and

$$\begin{aligned}
&\delta'(q'_0, a_1) = (q_{-n+2}, \ldots, q_1, a_{-n+3}, \ldots, a_1),\\
&\delta'((q_{-n+2}, \ldots, q_1, a_{-n+3}, \ldots, a_1), a_2) = (q_{-n+3}, \ldots, q_2, a_{-n+4}, \ldots, a_2),\\
&\qquad\qquad\qquad\qquad\vdots\\
&\delta'((q_{m-n}, \ldots, q_{m-1}, a_{m-n+1}, \ldots, a_{m-1}), a_m) =\\
&\qquad\qquad\qquad\qquad\qquad\qquad (q_{m-n+1}, \ldots, q_m, a_{m-n+2}, \ldots, a_m).
\end{aligned}$$

We will show by induction that for every prefix v of t of size at least n, there must exist some word $w \in \varrho^*(L)$ with v as infix. It is true for $a_1\cdots a_n$ because $f(q_0, a_1, \ldots, a_n) \neq \emptyset$ since $\delta'((q_0, \ldots, q_{n-1}, a_1, \ldots, a_{n-1}), a_n)$ is defined, which implies that $\inf(T_u) \cap a_1\cdots a_n\Sigma^* \neq \emptyset$ and hence $a_1\cdots a_n \in \inf(T_u)$ which is included in $\inf(\varrho^*(L))$, by Lemma 4.1. Let k be an integer such that $n \leq k < m$. Assume, by way of induction, that $a_1a_2\cdots a_k \leq_{\mathrm{i}} w \in \varrho^*(L)$, for some w. Since

$$\begin{aligned}
&\delta'((q_{k-n+1}, \ldots, q_k, a_{k-n+2}, \ldots, a_k), a_{k+1}) \;=\\
&\qquad\qquad\qquad\qquad (q_{k-n+2}, \ldots, q_{k+1}, a_{k-n+3}, \ldots, a_{k+1}),
\end{aligned}$$

it follows that $f(q_{k-n+1}, a_{k-n+2}, \ldots, a_{k+1}) \neq \emptyset$. Thus, there exists some $v \in \inf(T_u)$ with $v \in a_{k-n+2}\cdots a_{k+1}\Sigma^*$ and $\delta(q_{k-n+1}, v) \in F$. It follows that $a_1a_2\cdots a_{k-n+1}v = a_1a_2\cdots a_{k-n+1}a_{k-n+2}\cdots a_{k+1}\acute{v} \in R$ since[2] we know

[2]As in Theorem 4.1, we let $\acute{v} = \{\Sigma^n\}^{-1}\{v\}$, for $v \in \Sigma^*$.

that both $\delta(q_0, a_1a_2\cdots a_{k-n+1}) = q_{k-n+1}$ and $\delta(q_{k-n+1}, v) \in F$. Also, since $v \in \inf(T_u)$, there must exist $r_1, r_2, s_1, s_2 \in \Sigma^*$ such that $r_1vr_2 \in T_u$ and $s_1r_1vr_2s_2 \in \varrho^*(L)$ by Lemma 4.1. Also, there must exist $d_1, d_2 \in \Sigma^*$ such that $w = d_1a_1a_2\cdots a_kd_2 \in \varrho^*(L)$ by the inductive hypothesis. Furthermore,

$$(d_1a_1a_2\cdots a_kd_2, s_1r_1a_{k-n+2}\cdots a_{k+1}\acute{v}r_2s_2) \vdash_{a_1a_2\cdots a_{k+1}\acute{v}} d_1a_1a_2\cdots a_ka_{k+1}\acute{v}r_2s_2$$

since $k+1 \geq 2n_1 + n_2, a_1\cdots a_{k+1}\acute{v} = \alpha\beta\gamma$ with $\gamma = \overbrace{a_{k-n_1+2}\cdots a_{k+1}}^{n_1}\acute{v}, \beta = \overbrace{a_{k-n_1-n_2+2}\cdots a_{k-n_1+1}}^{n_2}, \alpha = \overbrace{a_1\cdots a_{k-n_1-n_2+1}}^{\geq n_1}$ and thus

$$a_1a_2\cdots a_ka_{k+1} \leq_{\mathrm{i}} d_1a_1a_2\cdots a_ka_{k+1}\acute{v}r_2s_2 \in \varrho^*(L).$$

Hence, by induction, for every l, $n \leq l \leq m$, $a_1a_2\cdots a_l \leq_{\mathrm{i}} w_l \in \varrho^*(L)$, for some w_l. Thus, $a_1a_2\cdots a_m \leq_{\mathrm{i}} w_m \in \varrho^*(L)$, $a_1a_2\cdots a_m \in T$ and $a_1a_2\cdots a_m$ is useful by Lemma 4.1. Thus, $t \in T_u$.

Hence, $T_u = T'$ and the useful subset of T on L is a regular language. ■

We denote the family of regular languages by **REG**. We can combine Theorem 4.2, Theorem 4.1, Lemma 3.1 and the facts that **REG** is the smallest full AFL and $\varrho^*(L) = \varrho_u^*(L)$ for every TGR system ϱ and language L, to obtain the following result:

Theorem 4.3 *Let $\mathcal{L}$ be a full AFL. Then*[3]

$$\pitchfork^*(\mathcal{L}, \mathbf{REG}) = \mathcal{L}.$$

Despite the fact that this proof does not provide an effective construction, it still exposes some necessary patterns that must occur after applying iterated template-guided recombination and sheds light on its (lack of) computational power.

5 Conclusions

We have continued the work of [2] and presented further formal studies of iterated template-guided recombination of DNA in stichotrichous ciliates,

[3]This theorem is also not effective.

recently proposed in [13]. Specifically, we introduced the notion of a template word and language being useful. We used this notion to show a type of "pumping lemma". This is used to demonstrate that $n_1 + n_2$ symbols at the left end, $a_1 \cdots a_{n_1+n_2}$ say, and right end, $b_1 \cdots b_{n_1+n_2}$ say, of each useful template are enough to guide insertion of the segment between any strand containing $a_1 \cdots a_{n_1+n_2}$ as subsequence and any strand containing $b_1 \cdots b_{n_1+n_2}$ as subsequence. This shows that if n_1 and n_2 are too small, then useful template words can be inserted, at random, quite frequently.

We also used this pumping lemma as a type of generative device, providing a characterization of iterated template-guided recombination. We then used this characterization to show that every full AFL $\mathcal{L}$ is closed under iterated template-guided recombination using useful templates also from $\mathcal{L}$.

We then presented a proof (which doesn't provide an effective construction) that the useful subset of a regular template language on an arbitrary initial language (without any restrictions) must also be regular. Consequently, this shows that every full AFL is closed under iterated template-guided recombination with regular template languages.

There are still many important open questions to be solved. First, from a bioinformatical point of view, is the pumping lemma above consistent with experimental evidence? In addition, from a computational standpoint, can one *effectively* find the useful subset of a regular template language (depending on the initial language family), or other template languages beyond the family of regular languages? This would also lend itself to deciding whether a given string w, a template language T and an initial language L satisfy $w \in \varrho^*(L)$. Other questions of interest would be whether it is possible to make this model capable of universal computation with finite or regular languages and small modifications based on biologically realistic assumptions.

The ultimate goal of this research is to both obtain a better, more formal understanding of ciliate genetics and to provide an elegant model of natural computing with the potential to be harnessed to perform difficult computations.

6 Acknowledgments

We thank Joost Engelfriet for helpful suggestions improving the presentation of this paper.

References

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

[2] M. Daley and I. McQuillan. Template-guided DNA recombination. *Theoretical Computer Science*, 330(2):237–250, 2005.

[3] A. Ehrenfeucht, T. Harju, I. Petre, D.M. Prescott, and G. Rozenberg. *Computation in Living Cells, Gene Assembly in Ciliates*. Springer-Verlag, Berlin, 2004.

[4] A. Ehrenfeucht, T. Harju, I. Petre, and G. Rozenberg. Patterns of micronuclear genes in ciliates. In N. Jonoska and N. Seeman, editors, *DNA7, Lecture Notes in Computer Science*, volume 2340, pages 279–289. Springer-Verlag, 2002.

[5] A. Ehrenfeucht, D.M. Prescott, and G. Rozenberg. Computational aspects of gene (un)scrambling in ciliates. In L.F. Landweber and E. Winfree, editors, *Evolution as Computation*, pages 45–86. Springer-Verlag, Berlin, Heidelberg, 2001.

[6] A. Ehrenfeucht, D.M. Prescott, and G. Rozenberg. Molecular operations for DNA processing in hypotrichous ciliates. *European Journal of Protistology*, 37(3):241–260, 2001.

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

[8] L. Kari and L.F. Landweber. Computational power of gene rearrangement. In E. Winfree and D. Gifford, editors, *DNA5, DIMACS series in Discrete Mathematics and Theoretical Computer Science*, volume 54, pages 207–216. American Mathematical Society, 2000.

[9] L.F. Landweber and L. Kari. The evolution of cellular computing: Nature's solution to a computational problem. In L. Kari, H. Rubin, and D.H. Wood, editors, *DNA4, BioSystems*, volume 52, pages 3–13. Elsevier, 1999.

[10] D.M. Prescott. Cutting, splicing, reordering, and elimination of DNA sequences in hypotrichous ciliates. *BioEssays*, 14(5):317–324, 1992.

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

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

[13] D.M. Prescott, A. Ehrenfeucht, and G. Rozenberg. Template-guided recombination for IES elimination and unscrambling of genes in stichotrichous ciliates. *Journal of Theoretical Biology*, 222(3):323–330, 2003.

[14] A. Salomaa. *Formal Languages*. Academic Press, New York, 1973.