# On Computational Properties of Template-Guided DNA Recombination*

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

[1] Department of Computer Science and Department of Biology,
University of Western Ontario,
London, Ontario, N6A 5B7, Canada
daley@csd.uwo.ca
[2] Department of Computer Science,
University of Saskatchewan,
Saskatoon, Saskatchewan, S7N 5A9, Canada
imcquill@csd.uwo.ca

**Abstract.** The stichotrichous ciliates have attracted the attention of both biologists and computer scientists due to the unique genetic mechanism of gene descrambling. It has been suggested that it would perhaps be possible to co-opt this genetic process and use it to perform arbitrary computations *in vivo*. Motivated by this idea, we study here some basic properties and the computational power of a formalization inspired by the template-guided recombination model of gene descrambling proposed by Ehrenfeucht, Prescott and Rozenberg. We demonstrate that the computational power of a system based on template-guided recombination is quite limited. We then extend template-guided recombination systems with the addition of "deletion contexts" and show that such systems have strictly greater computational power than splicing systems [1, 2].

## 1 Introduction

The stichotrichous ciliates are a family of single-celled organisms that have come to be studied by both biologists and computer scientists due to the curious mechanism of gene scrambling. Every stichotrichous ciliate has both a functional macronucleus, which performs the "day-to-day" genetic chores of the cell, and an inert micronucleus. Although stichotrichs reproduce asexually, they do also conjugate to exchange genetic material. This hopefully increases the genetic diversity and strength of both organisms involved in conjugation.

The micronucleus contains germline DNA which becomes important during the process of conjugation between two cells. 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.

* This research was funded in part by institutional grants of the University of Saskatchewan and the University of Western Ontario, the SHARCNET Research Chairs programme, the Natural Sciences and Engineering Research Council of Canada and the National Science Foundation of the United States.

A. Carbone and N.A. Pierce (Eds.): DNA11, LNCS 3892, pp. 27–37, 2006.

The interest in this process from a computational point of view comes from the fact that the genes in the micronucleus are stored in a scrambled order. Specifically, the micronuclear gene consists of fragments of the macronuclear gene in some permuted order. That is, if we denote a functional macronuclear gene with the string "1-2-3-4-5", then the equivalent gene in the micronucleus may appear as "2-$\epsilon_1$-4- $\epsilon_2$-1-$\epsilon_3$-3-$\epsilon_4$-5", where the $\epsilon$'s represent so-called internally eliminated sequences (or IES's) which are removed from the macronuclear version of the gene. Each sequence, 1 through 5, is referred to as a macronuclear destined sequence (or MDS).

The cell must thus have some mechanism to de-scramble these fragments in order to create a functional gene which is capable of generating a protein. For more information on the biological process of gene de-scrambling, we refer to [3].

Several models for how this de-scrambling process takes place have been proposed in the literature. There are two primary theoretical models which have been investigated: the Kari-Landweber model [4, 5] which consists of a binary inter- and intra-molecular recombination operation and the Ehrenfeucht, Harju, Petre, Prescott and Rozenberg model [6, 7, 8] which consists of three unary operations inspired by intramolecular DNA recombination.

Recently, a new model has been proposed by Prescott, Ehrenfeucht and Rozenberg [9] based on the recombination of DNA strands guided by templates.

The basic action of the model is to take two DNA segments and splice them together via a template intermediary, if the form of the segments matches the form of the template. Consider DNA segments of the form $u\alpha\beta d$ and $e\beta\gamma v$ where $u, v, \alpha, \beta, d, e, \gamma$ are subsequences of a DNA strand. If we wish to splice these two strands 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 $\alpha$ 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 according to the biochemistry of DNA and we are left 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: $u\alpha\beta\gamma v$. For more details on this operation, we refer to [9].

It has been suggested that the *in vivo* computational process of gene descrambling may be able to be controlled in such a way that it would be possible to perform an arbitrary computation with a ciliate. Taking this as our motivation, in this paper we present a generalized version of the template-guided recombination operation and study the basic properties and computational power of both non-iterated and iterated versions. We conclude that, even in the iterated case, the computational power is quite limited and propose a straightforward extension to a model which is strictly more computationally powerful than splicing systems.

The paper is organized as follows; Section 2 of the paper will present formal language theoretic prerequisites and notation. In section 3 we consider the basic closure properties and the computational power of the template-guided recombination operation. We then contrast this by recalling results on an iterated version of this operation. The limited computational power of both the iterated

and non-iterated versions leads us to study a context-aware extension of the operation, which proves strictly more powerful than splicing systems, in Section 4. We present our conclusions in section 5.

## 2 Preliminaries

We refer to [10] for language theory preliminaries. Let $\Sigma$ be a finite alphabet. We denote, by $\Sigma^*$ and $\Sigma^+$, the sets of all words and non-empty words, respectively, over $\Sigma$ and the empty word by $\lambda$. A language $L$ is any subset of $\Sigma^*$. Let $x \in \Sigma^*$. We let $|x|$ denote the length of $x$. For $n \in \mathbb{N}_0$, let $\Sigma^{\leq n} = \{w \in \Sigma^* \mid |w| \leq n\}$, $\Sigma^{\geq n} = \{w \in \Sigma^* \mid |w| \geq n\}$ and $\Sigma^n = \{w \in \Sigma^* \mid |w| = n\}$. A homomorphism $h : X^* \to Y^*$ is termed a coding if $|h(a)| = 1$ for each $a \in X$ and $h$ is termed a weak coding if $|h(a)| \leq 1$ for each $a \in X$. 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 $LR^{-1} = \{z \in \Sigma^* \mid zy \in L \text{ for some } y \in R\}$.

We denote the family of finite languages by **FIN**, regular languages by **REG**, linear languages by **LIN**, context-free languages by **CF**, context-sensitive languages by **CS** and recursively enumerable languages by **RE**.

A *trio* is a non-trivial language family closed under $\lambda$-free homomorphism, inverse homomorphism and intersection with regular sets. It is known that every trio is closed under $\lambda$-free $a$-transductions[1] and inverse gsm mappings. An *AFL* is a trio closed under arbitrary union, concatenation and $+$. A *full trio*[2] 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. A *full semi-AFL* is a full trio closed under union. A *full AFL* is a full trio closed under arbitrary union, concatenation and Kleene $*$. It can be seen that **REG**, **CF** and **RE** are full AFL's, **LIN** is a full semi-AFL not closed under concatenation or $*$, and **CS** is an AFL not closed under arbitrary homomorphism. We refer to [11, 12] for the theory of AFL's.

## 3 Template-Guided Recombination

We will first formally define the template-guided recombination operation as it appears in [13, 14].

**Definition 1.** *A template-guided recombination system (or* TGR *system) is a four tuple $\varrho = (T, \Sigma, n_1, n_2)$ where $\Sigma$ 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 $\varrho(L) = \{w \in \Sigma^* \mid (x, y) \vdash_t w \text{ for some } x, y \in L, t \in T\}$ where $(x, y) \vdash_t w$ if and only if $x = u\alpha\beta d, y = e\beta\gamma v, t = \alpha\beta\gamma, w = u\alpha\beta\gamma v, u, v, d, e \in \Sigma^*, \alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{\geq n_2}$.*

[1] An $a$-transducer is also referred to as a rational transducer.

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

*Let* $\mathcal{L}_1, \mathcal{L}_2$ *be language families and* $n_1, n_2 \in \mathbb{N}$. *We write* $\pitchfork(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) = \{\varrho(L) \mid L \in \mathcal{L}_1, \varrho = (T, \Sigma, n_1, n_2)$ *a* TGR *system*, $T \in \mathcal{L}_2\}$ *and* $\pitchfork(\mathcal{L}_1, \mathcal{L}_2) = \bigcup_{n_1,n_2 \in \mathbb{N}} \pitchfork(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2)$

We remark here that while the operation of template-guided recombination bears a superficial resemblance to the splicing operation introduced by Head [1] and extended by Paun, et. al. [2], the operations are, in fact, distinct. While TGR systems are in most cases less computationally powerful than comparable splicing systems, they are often more succinct in terms of the descriptional complexity of a system generating a particular language. Moreover, we will show in this paper that a contextual extension of TGR systems is strictly more computationally powerful than the inherently contextual splicing systems. Further details on the relationship between splicing systems and TGR systems can be found in [13].

*Remark 1.* In [9], a constant $C$ is defined such that $|\alpha|, |\gamma| > C$ in order to ensure the formation of sufficiently strong chemical bonds. Likewise, [9] also defines constants $D$ and $E$ such that $D < |\beta| < E$. The definition, as above, and also the results in this paper, are general enough to cover any such $D$ and $C$. In addition, the constant $E$, as defined above, is shown to be irrelevant in the next proposition. It was noted in [9] that the smallest pointer sequence known was of length 3, although recently, a pointer sequence was discovered experimentally which was only of length one [15]. Also, we believe that the smallest MDS sequence discovered to date is nine nucleotides long [16]. In any case, the notation above is general enough to work for any such constants. We also note that the notation above will work when the two operands $x$ and $y$ in Definition 1 are either the same or when they are not. It has been seen experimentally that two MDS's can be on two different loci but still recombine successfully.

The following proposition, from [13] states that we can always assume that the $\beta$ subword of a template is of the minimum length, $n_2$.

**Proposition 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* $x = u\alpha\beta d, y = e\beta\gamma v, t = \alpha\beta\gamma, w = u\alpha\beta\gamma v, u, v, d, e \in \Sigma^*, \alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{n_2}$.

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

We now consider new results regarding the power of template-guided recombination when restricted to a single application of the operation. This is important to the basic theoretical understanding of how the operation functions relative to traditional theoretical computer science. We omit proofs here due to space considerations.

First, we show that, under some weak restrictions, closure under intersection follows from closure under template-guided recombination.

**Lemma 1.** *Let* $\mathcal{L}_1$ *be a language family closed under left and right concatenation and quotient with a single symbol and under union with singleton languages and*

*let $\mathcal{L}_2$ be a language family closed under left and right concatenation with a single symbol such that $\pitchfork(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) \subseteq \mathcal{L}_1$ for some $n_1, n_2 \in \mathbb{N}$. The intersection of a language from $\mathcal{L}_1$ with a language from $\mathcal{L}_2$ belongs to $\mathcal{L}_1$.*

Since $\Sigma^*$ is in every language family containing **REG** and $\Sigma^* \cap T = T$, we obtain:

**Corollary 1.** *Let $\mathcal{L}_1$ be a language family such that* **REG** $\subseteq \mathcal{L}_1$*, $\mathcal{L}_1$ is closed under left and right concatenation and quotient with a symbol and union with singleton languages and let $\mathcal{L}_2$ be a language family closed under left and right concatenation with a symbol such that $\pitchfork(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) \subseteq \mathcal{L}_1$ for some $n_1, n_2 \in \mathbb{N}$. Then $\mathcal{L}_2 \subseteq \mathcal{L}_1$.*

We now continue the characterization of template-guided recombination in terms of AFL theory. We see that, under some restrictions, closure under concatenation follows from closure under template-guided recombination.

**Lemma 2.** *Let $\mathcal{L}_1$ be a language family closed under limited erasing homomorphism, union, left and right concatenation by a symbol and let $\mathcal{L}_2$ be a language family containing the singleton languages such that $\pitchfork(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) \subseteq \mathcal{L}_1$ for some $n_1, n_2 \in \mathbb{N}$. Then $\mathcal{L}_1$ is closed under concatenation.*

We now show that we can simulate template-guided recombination with a few standard operations.

**Lemma 3.** *Let $\mathcal{L}_1$ be closed under marked concatenation*[3]*, intersection with regular languages and inverse gsm mappings. Let $\mathcal{L}_2$ be closed under inverse gsm mappings and intersection with regular languages. Let $L \in \mathcal{L}_1, T \in \mathcal{L}_2$ and let $\varrho = (T, \Sigma, n_1, n_2)$ be a* TGR *system. Then there exists $L' \in \mathcal{L}_1, T' \in \mathcal{L}_2$ and a weak coding homomorphism $h$ such that $\varrho(L) = h(L' \cap T')$.*

Since every trio is closed under inverse gsm mappings, we get the following:

**Corollary 2.** *Let $\mathcal{L}_1$ be a concatenation closed full trio and let $\mathcal{L}_2$ be either a trio or $\mathcal{L}_2 \subseteq$* **REG***. If $\mathcal{L}_1$ is closed under intersection with $\mathcal{L}_2$ then $\pitchfork(\mathcal{L}_1, \mathcal{L}_2) \subseteq \mathcal{L}_1$.*

We combine the lemmata above to obtain the following result:

**Proposition 2.** *Let $\mathcal{L}_1$ be a full semi-AFL and $\mathcal{L}_2$ be a trio or $\mathcal{L}_2 =$* **FIN***. Then $\pitchfork(\mathcal{L}_1, \mathcal{L}_2) \subseteq \mathcal{L}_1$ if and only if $\mathcal{L}_1$ is closed under concatenation and $\mathcal{L}_1$ is closed under intersection with $\mathcal{L}_2$.*

Since every full semi-AFL is closed under intersection with regular languages, it now follows that for a full semi-AFL, closure under catenation is necessary and sufficient to show closure under template-guided recombination with regular and finite languages.

**Corollary 3.** *For every full semi-AFL $\mathcal{L}$, $\pitchfork(\mathcal{L},$* **REG***$) \subseteq \mathcal{L}$ and $\pitchfork(\mathcal{L},$* **FIN***$) \subseteq \mathcal{L}$ if and only if it is closed under concatenation.*

[3] The marked concatenation of $L_1, L_2$ is $L_1 a L_2$ where $a$ is a new symbol.

Likewise, the next result concerning the closure of intersection-closed full semi-AFLs now follows since every intersection-closed full semi-AFL is closed under concatenation.

**Corollary 4.** *For every intersection-closed full semi-AFL* $\mathcal{L}$, $\pitchfork(\mathcal{L}, \mathcal{L}) \subseteq \mathcal{L}$.

The above results are sufficient to characterize the closure properties of the families of finite, regular, linear and context-free families. We now show that the family of context-sensitive languages is not even closed under template-guided recombination with singleton languages.

**Proposition 3.** $\pitchfork(\mathbf{CS}, \mathbf{FIN}, n_1, n_2) \not\subseteq \mathbf{CS}$ *for any* $n_1, n_2 \in \mathbb{N}$.

Now, we can completely fill in a table (see Table 1) with the families of languages in the Chomsky hierarchy, the finite languages and the linear languages. A $\surd$ represents closure of $\mathcal{L}_1$ under template-guided recombination with templates from $\mathcal{L}_2$ and a blank represents non-closure. The results hold for any minimum pointer and MDS length.

**Table 1.** $\pitchfork(\mathcal{L}_1, \mathcal{L}_2) \subseteq \mathcal{L}_1$?

| $\mathcal{L}_1$ \| $\mathcal{L}_2$ | **FIN** | **REG** | **LIN** | **CF** | **CS** | **RE** |
|---|---|---|---|---|---|---|
| **FIN** | $\surd$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ |
| **REG** | $\surd$ | $\surd$ | | | | |
| **LIN** | | | | | | |
| **CF** | $\surd$ | $\surd$ | | | | |
| **CS** | | | | | | |
| **RE** | $\surd$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ |

In a biological system it is natural to investigate iterated application of operations as bio-operations are the product of the stochastic biochemical reactions of enzymes, catalysts and substrates in solution. We now recall results on an iterated version of the template-guided recombination operation.

We begin with the definition iterated template-guided recombination from [14]:

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

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

Let $\mathcal{L}_1, \mathcal{L}_2$ be language families and $n_1, n_2 \in \mathbb{N}$. We define $\pitchfork^*(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2) = \{\varrho^*(L) \mid L \in \mathcal{L}_1, \varrho = (T, \Sigma, n_1, n_2)$ a TGR system, $T \in \mathcal{L}_2\}$ and let $\pitchfork^*(\mathcal{L}_1, \mathcal{L}_2) = \bigcup_{n_1, n_2 \in \mathbb{N}} \pitchfork^*(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2)$.

We now give a short example of an iterated template-guided recombination system.

*Example 1.* Let $L = \{a^{n_1}a^{n_2}\}$ and let $\varrho = (T, \Sigma, n_1, n_2)$ be a TGR system where $\Sigma = \{a\}$, $T = \{a^{n_1}a^{n_2}a^{n_1}\}$ and $n_1, n_2 \in \mathbb{N}$. Then $\varrho^*(L) = \{a^{n_1+n_2}\} \cup \{a^{2n_1+n_2}a^*\}$. For the case $n_1 = n_2 = 1$, $\varrho(L) = aa^+$.

It is also known from [13] that the closure of a language family under iterated template-guided recombination contains the original language family.

**Lemma 4.** *Let $\mathcal{L}_1, \mathcal{L}_2$ be language families and let $n_1, n_2 \in \mathbb{N}$. Then $\mathcal{L}_1 \subseteq \pitchfork^*(\mathcal{L}_1, \mathcal{L}_2, n_1, n_2)$.*

We have considered the basic properties of the iterated version of template-guided recombination in [14] and we recall here the definition of a useful template from that paper.

Intuitively, a template word is useful if it can be used as a template to produce any word, not necessarily new. The full formal definition is found in [14]. This notion turns out to be quite important as is shown by the following two results.

We see that every full AFL is closed under iterated template-guided recombination with useful templates from the same full AFL.

**Theorem 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 $\varrho$ is useful on $L$. Then $\varrho^*(L) \in \mathcal{L}$.*

In addition, when the template sets are regular, the useful subset, $T_u$ say, of the template language $T$ on *any* language $L$ has a very simple structure relative to $T$.

**Proposition 4.** *Let $\varrho = (T, \Sigma, n_1, n_2)$ be a* TGR *system, let $L \subseteq \Sigma^*$ and let $T_u$ be the useful subset of $T$ on $L$. If $T$ is a regular language, it follows that $T_u$ is also regular.*

The language $L$ in the proposition above does not have any restrictions placed on it. It need not even be recursively enumerable. The proof does not, however, provide an effective construction for $T_u$.

A consequence of Theorem 1, Proposition 4 and the fact that the family of regular languages is the smallest full AFL allows us to show the following key result.

**Theorem 2.** *Let $\mathcal{L}$ be a full AFL and let $n_1, n_2 \in \mathbb{N}$. Then*

$$\pitchfork^*(\mathcal{L}, \mathbf{REG}, n_1, n_2) = \mathcal{L}.$$

This shows that the operation, as defined, provides very little computational power, regardless of the minimum pointer and MDS length. Indeed, even when we start with regular initial and template languages, we cannot generate any non-regular languages. This is not surprising biologically, however, as one might expect the cell to make use of the least complex computational process to accomplish a given task.

In the next section we show that adding even a small amount of context-sensitiveness to template-guided recombination results in a large increase in computational power.

## 4 Extension of TGR by Deletion Contexts

As defined above, the operation of template-guided recombination is able to achieve very limited computational power. Even the iterated version is able only to generate regular languages starting from a regular initial language and using a regular set of templates. This is in contrast to the fact that extended splicing systems are able to generate arbitrary recursively enumerable languages starting from regular splicing rules and a finite set of axioms[17]. It is often the case that small alterations to an operation can lead to a huge increase in generative capacity. In this section, we add a feature to this operation in order to achieve more power. It should be stated immediately that it is not clear how realistic this extension is in a biological setting. While the extension presented is certainly not biologically impossible, neither do we have experimental evidence to support it. Despite this, it serves as an aide to the study of what properties should likely be present in order to obtain more general computation.

We begin by defining a more general version of the template-guided recombination operation. Indeed, the new notation allows for extra deletion contexts, beyond the $\beta$ pointer. The previously studied operation is a special case where all deletion contexts are of length zero. We cannot assume, with this more general notation that the symbol $\beta$ is always of the minimum pointer length.

**Definition 2.** *A contextual template-guided recombination system (or shortly, a* CTGR *system) is a four tuple $\varrho = (T, \Sigma, n_1, n_2)$ where $\Sigma$ is a finite alphabet, $\#$ is a symbol not in $\Sigma$, $T \subseteq \Sigma^*\#\Sigma^*\#\Sigma^*$ 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* CTGR *system $\varrho = (T, \Sigma, n_1, n_2)$ and a language $L \subseteq \Sigma^*$, we define $\varrho(L) = \{w \in \Sigma^* \mid (x, y) \vdash_t^c w$ for some $x, y \in L, t \in T\}$ where $(x, y) \vdash_t^c w$ if and only if $x = u\alpha\beta d_1 d, y = ee_1\beta\gamma v, t = e_1\#\alpha\beta\gamma\#d_1, w = u\alpha\beta\gamma v, u, v, d, e \in \Sigma^*, \alpha, \gamma \in \Sigma^{\geq n_1}, \beta \in \Sigma^{\geq n_2}$.*

*For $k \in \mathbb{N}_0 \cup \{\infty\}$, we denote $\underline{\Uparrow}(\mathcal{L}_1, \mathcal{L}_2[k], n_1, n_2) = \{\varrho(L) \mid L \in \mathcal{L}_1, \varrho = (T, \Sigma, n_1, n_2)$ a* CTGR *system, $T \in \mathcal{L}_2, T \subseteq \Sigma^{\leq k}\#\Sigma^*\#\Sigma^{\leq k}\}$ and $\underline{\Uparrow}(\mathcal{L}_1, \mathcal{L}_2) = \{\underline{\Uparrow}(\mathcal{L}_1, \mathcal{L}_2[\infty], n_1, n_2) \mid n_1, n_2 \in \mathbb{N}\}$.*

We then get template-guided recombination as a special case where the contexts are of length zero. Next, we see that if we add in even one symbol of deletion context, we increase the power significantly.

**Lemma 5.** *Let $\Sigma$ be an alphabet, $\Sigma_1 = \Sigma \cup \{a_1, a_2, a_3, a_4, a_5\}$, (all new symbols disjoint from $\Sigma$), $\mathcal{L}$ a language family closed under left and right concatenation with symbols and $L_1 \cup aL_2 \in \mathcal{L}$ for $L_1, L_2 \in \mathcal{L}$, a a new symbol. Then there exists $L \in \mathcal{L}, T \in \Sigma_1\#\Sigma_1^*\#\Sigma_1$, $T \in \mathbf{REG}_0$ and a* CTGR *system $\varrho = (T, \Sigma_1, 1, 1)$ such that $L_1 \cap L_2 = (a_4a_1a_2)^{-1}(\varrho(L))(a_3a_1a_5)^{-1}$ and $\varrho(L) \in \underline{\Uparrow}(\mathcal{L}, \mathbf{REG}_0[1], 1, 1)$.*

As corollary, we obtain that $\underline{\Uparrow}(\mathbf{LIN}, \mathbf{REG}_0[1], 1, 1)$ is equal to $\mathbf{RE}$ after applying an intersection with a regular language and a homomorphism.

**Corollary 5.** *Let $\Sigma$ be an alphabet, $L \in \mathbf{RE}, L \subseteq \Sigma^*$. Then there exists an alphabet $\Sigma_1$, a homomorphism $h$ from $\Sigma_1^*$ to $\Sigma^*$, languages $R, T \in \mathbf{REG}_0$, a language $L' \in \mathbf{LIN}$ and a* CTGR *system $\varrho = (T, \Sigma_1, 1, 1)$ such that $h(\varrho(L') \cap R) = L$.*

Thus, even though $\pitchfork(\mathcal{L}_1, \mathbf{REG}) \subseteq S(\mathcal{L}_1, \mathbf{REG})$[4] for every $\mathcal{L}_1$ (see [13]) and $S(\mathcal{L}_1, \mathbf{REG}) \subseteq \mathcal{L}_1$ for every concatenation closed full trio $\mathcal{L}_1$ (see [2]), we see that when we add in even one symbol of deletion contexts, $\underline{\pitchfork}(\mathcal{L}_1, \mathbf{REG}) \not\subseteq S(\mathcal{L}_1, \mathbf{REG})$ in many cases, for example when $\mathcal{L}_1$ is the family of context-free languages. Consequently, template-guided recombination with deletion contexts can generate more powerful languages than splicing systems.

**Lemma 6.** *Let $\mathcal{L}_1, \mathcal{L}_2$ be language families, both closed under inverse gsm mappings and intersection with regular languages, let $L \in \mathcal{L}_1, T \in \mathcal{L}_2$ and let $\varrho = (T, \Sigma, n_1, n_2)$ be a* CTGR *system. Then there exists $L_1, L_2 \in \mathcal{L}_1, T' \in \mathcal{L}_2$ and a weak coding homomorphism $h$ such that $\varrho(L) = h(L_1 \cap L_2 \cap T')$.*

**Corollary 6.** *Let $\mathcal{L}_1$ be an intersection-closed full trio closed under intersection with $\mathcal{L}_2$, which is either closed under inverse gsm mappings and intersection with regular languages or $\mathcal{L}_2 \subseteq \mathbf{REG}$. Then $\underline{\pitchfork}(\mathcal{L}_1, \mathcal{L}_2) \subseteq \mathcal{L}_1$.*

**Proposition 5.** *Let $\mathcal{L}_1$ be a full semi-AFL and let $\mathbf{REG}_0 \subseteq \mathcal{L}_2$ be closed under inverse gsm mappings and intersection with regular languages. Then $\underline{\pitchfork}(\mathcal{L}_1, \mathcal{L}_2) \subseteq \mathcal{L}_1$ if and only if $\mathcal{L}_1$ is closed under intersection with $\mathcal{L}_1$ and $\mathcal{L}_2$.*

We would also like to study the iterated version of this more general operation. Let $\varrho = (T, \Sigma, n_1, n_2)$ be a CTGR system and let $L \subseteq \Sigma^*, T \subseteq \Sigma^* \# \Sigma^* \# \Sigma^*$. We generalize $\varrho$ to an iterated operation $\varrho^*(L)$ in the natural way:

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

In the following, we show that we are able to generate arbitrary recursively enumerable languages using iterated contextual template-guided recombination with regular templates and a finite initial language and applying an intersection with a terminal alphabet and a coding. The following proof follows the well known "simulate-rotate" proof technique from splicing systems [2]. We apply the final coding homomorphism in the proof in order to stop the $\beta$ symbol in the definition from "compressing" small amounts of information in an undesirable fashion. It is not clear if the coding is strictly necessary, however it is very simple: mapping three separate symbols onto one for each symbol. We only require deletion contexts of length two.

**Proposition 6.** *Let $L' \subseteq \Sigma^*$ be an arbitrary recursively enumerable language. Then there exist alphabets $\overline{\Sigma}, W$, a regular template language $T$, a* CTGR *system $\varrho = (T, W, 1, 1)$, a finite language $L \subseteq W^*$, and a coding homomorphism $h$ from $\overline{\Sigma}^*$ to $\Sigma^*$ such that $h(\varrho^*(L) \cap \overline{\Sigma}^*) = L'$.*

We have thus demonstrated that an arbitrary recursively enumerable language can be generated by a CTGR system with a finite initial language and a

[4] Where $S(\mathcal{L}_1, \mathbf{REG})$ denotes non-iterated splicing systems with an initial language in $\mathcal{L}_1$ and splicing rules in **REG**.

regular template language up to a coding homomorphism and intersection with a terminal alphabet.

## 5 Conclusions

We have considered the basic properties and computational power of an operation inspired by the template-guided recombination model of gene descrambling in stichotrichous ciliates. Specifically, we began by investigating the properties of a non-iterated version of template-guided recombination systems, contributing to their theoretical understanding. We characterized closure properties of families of languages under template-guided recombination in terms of other basic operations and demonstrated that every intersection-closed full semi-AFL is closed under template-guided recombination with templates from the same full semi-AFL.

We then recalled the properties of iterated template-guided recombination systems. The principal result here shows the limited power of template-guided recombination by demonstrating that every full AFL is closed under iterated template-guided recombination with regular templates. This implies that the computational power of any system based on this operation will be quite limited. Indeed, if one enforces the "reasonable" restriction that template and initial languages must be regular, one does not gain any increase in computational power. This motivates the question of what minimal extension would be required to increase the generative capacity beyond the regular languages while still restricting the initial and template languages to be, at most, regular.

We addressed this question by showing that the tight restriction on computational power can be lifted by adding a small degree of context-awareness to a TGR system. We have demonstrated that we are able to generate arbitrary recursively enumerable languages using contextual iterated template-guided recombination with regular templates, a finite initial language and applying an intersection with a terminal alphabet and a coding.

It may be preferable, from the point of view of biocomputing, to show a result which demonstrates the simple template-guided recombination systems to be capable of universal computation; however, from the point of view of judging the closeness of this formalization to the biological process which it models, the opposite may be true. Given that a cell has access to only finite resources, and has serious constraints on the length of time in which the descrambling process must be completed, it seems reasonable that the process must be relatively computationally simple.

We have also shown in this paper that by adding a small amount of context-awareness to a TGR system, we are able to easily generate arbitrary recursively enumerable languages. While this result may be more theoretically satisfying, we caution that the "deletion contexts" required to derive such a result are not present in the biological model given in [9], though they are not biologically impossible. Too little is currently known about the molecular biology of ciliates to make definitive statements; however, we feel that, in the context of formalizing a biological process, a result indicating limited computational power is perhaps preferable.

The simplicity and elegance of template-guided recombination combined with the ubiquity of template-mediated events in biological systems, shows that the operation warrants further investigation both as a possible model of a biological process and as a purely abstract operation.

## Acknowledgments

We thank Grzegorz Rozenberg for helpful discussions.

## References

1. Head, T.: Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. Bulletin of Mathematical Biology **49** (1987)
2. Păun, G., Rozenberg, G., Salomaa, A.: DNA Computing : new computing paradigms. Springer-Verlag, Berlin (1998)
3. Prescott, D.: Genome gymnastics: Unique modes of DNA evolution and processing in ciliates. Nature Reviews Genetics **1** (2000) 191–198
4. Kari, L., Landweber, L.: Computational power of gene rearrangement. In Winfree, E., Gifford, D., eds.: DNA5, DIMACS series in Discrete Mathematics and Theoretical Computer Science. Volume 54. American Mathematical Society (2000) 207–216
5. Landweber, L., Kari, L.: The evolution of cellular computing: Nature's solution to a computational problem. In Kari, L., Rubin, H., Wood, D., eds.: DNA4, BioSystems. Volume 52. Elsevier (1999) 3–13
6. Ehrenfeucht, A., Prescott, D., Rozenberg, G.: Computational aspects of gene (un)scrambling in ciliates. In Landweber, L., Winfree, E., eds.: Evolution as Computation. Springer-Verlag, Berlin, Heidelberg (2001) 45–86
7. Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D., Rozenberg, G.: Computation in Living Cells, Gene Assembly in Ciliates. Springer-Verlag, Berlin (2004)
8. Ehrenfeucht, A., Prescott, D., Rozenberg, G.: Molecular operations for DNA processing in hypotrichous ciliates. European Journal of Protistology **37** (2001) 241–260
9. Prescott, D., Ehrenfeucht, A., Rozenberg, G.: Template-guided recombination for IES elimination and unscrambling of genes in stichotrichous ciliates. Journal of Theoretical Biology **222** (2003) 323–330
10. Salomaa, A.: Formal Languages. Academic Press, New York (1973)
11. Berstel, J.: Transductions and Context-Free Languages. B.B. Teubner, Stuttgart (1979)
12. Ginsburg, S.: Algebraic and Automata-Theoretic Properties of Formal Languages. North-Holland Publishing Company, Amsterdam (1975)
13. Daley, M., McQuillan, I.: Template-guided DNA recombination. Theoretical Computer Science **330** (2005) 237–250
14. Daley, M., McQuillan, I.: Useful templates and template-guided DNA recombination. (to appear in *Theory of Computing Systems*)
15. Doak, T.: (Personal communication)
16. Landweber, L., Kuo, T., Curtis, E.: Evolution and assembly of an extremely scrambled gene. PNAS **97** (2000) 3298–3303
17. Păun, G.: Regular extended H systems are computationally universal. Journal of Automata, Languages and Combinatorics **1** (1996) 27–36