# CUT, PASTE AND FILTER

PEGGY SULLIVAN

Abstract. We study the combined and separate effects of three parts of finite multi-test-tube cut and paste DNA computing. First, we reformulate the ideas of Csuhaj-Varjú, Kari, and Păun [1], Freund [3], and Priese, Rogojine and Margenstern [10] about multi-test-tube splicing DNA computing in terms of cutting and pasting as in Pixton's work. Pixton shows [8, 9] that with finite cutting and pasting only regular sets can be obtained from a finite set of initial molecules. The others listed above show that using filtering between a finite number of test-tubes, each with finite splicing, any recursively enumerable set can be obtained from finite initial contents [1, 3, 10]. We confirm their result for cutting and pasting. Second, we show that when only finite pasting and filtering between tubes with finite initial contents are allowed then the result must be context free and that any context free language can be so obtained. Finally, we consider several forms of filtering and several ways of combining filtering with cutting, pasting or splicing and show that all give equivalent results.

## 1. Basic Definitions

An *alphabet* $A$ is a non-empty finite set. $A^*$ is the free monoid generated by $A$. Its elements are called *words or strings (on A)*, and the identity of $A^*$, the empty word, is written 1. A subset of $A^*$ is a *language (on A)*. If $L$ is a language on $A$, let the *set of factors or sub-words of* $L$ be $\operatorname{Fac} L = \{w \in A^* \mid uwv \in L \text{ for some } u, v \in A^*\}$. If $L$ is a language on $A$ and $M$ is a language on $B$ then the result of concatenating them is $LM = \{w \in (A \cup B)^* \mid w = w_1 w_2 \text{ for some } w_1 \in L, w_2 \in M\}$, a language on $A \cup B$.

If $\Sigma$ is another finite non-empty set, of *end-markers*, then an element of $\Sigma A^* \Sigma$ is called an *end-marked string (on A with end-markers in Σ)* and a subset of $\Sigma A^* \Sigma$ is called an *end-marked language.* If $M$ is an end-marked language on $A$ with end-markers in $\Sigma$ then $\operatorname{Strip} M = \{w \in A^* \mid \alpha w \beta \in M \,\text{for some}\, \alpha, \beta \in \Sigma\}$, the *stripped version of* $M$, is the language obtained by removing the end-markers from the words of $M$. As a convention, we will use Roman letters for elements of $A$ and $A^*$ and Greek letters for elements of $\Sigma$. For multi-test-tube computing we will need tuples of end-marked languages. Let $\boldsymbol{L} = (L_1, L_2, \ldots, L_n)$ and $\boldsymbol{M} = (M_1, M_2, \ldots, M_n)$ be $n$-tuples of end-marked languages. We say $\boldsymbol{L} \subset \boldsymbol{M}$ provided $L_i \subset M_i$ for $1 \leq i \leq n$. Further we define $\boldsymbol{L} \cup \boldsymbol{M}$ component-wise by $(\boldsymbol{L} \cup \boldsymbol{M})_i = L_i \cup M_i$.

One way to generate languages is using phrase-structure grammars. The most general grammar we will use here is the *type-0 grammar* $G = (N, T, R, S)$. Here $N$ is a finite set of *non-terminal symbols*, $T$ is a finite set of *terminal symbols*, $R \subset (N \cup T)^* N^+ (N \cup T)^* \times (N \cup T)^*$ is a finite set of *production rules or productions*, and $S \in N$ is the *start symbol.* It is required that $N$ and $T$ be disjoint. If $(u, v) \in R$ we write $u \to v$. The relation $\to$ extends to a relation on $(N \cup T)^*$ as follows: for $x, y \in (N \cup T)^*$ and $u \to v$ in $R$ write $xuy \to xvy$. The relation $\xrightarrow{k}$ is the $k$-fold iteration of $\to$. The relation $\xrightarrow{*}$ is the reflexive, transitive closure

---

*Date*: July 7,2000.

*Key words and phrases.* DNA computing, multi-test-tube systems, CF languages.

of $\rightarrow$. We say $w \in T^*$ is in $L(G)$, *the language generated by* $G$, provided that $S \xrightarrow{*} w$. The family of languages generated by type-0 grammars is the family of *recursively enumerable (RE) languages.* If more restrictions are placed on the set $R$, other families in the Chomsky hierarchy of languages are generated. In particular, if we require that $R \subset N \times (N \cup T)^*$ then we have a *type-2 grammar* and such grammars generate precisely the *context-free (CF) languages.* There are many different normal forms for context-free grammars. We will use Chomsky normal form, in which every production is of the form $u \rightarrow vw$ or $u \rightarrow a$ or $u \rightarrow 1$ where $u, v, w \in N$ and $a \in T$.

A *CPH-scheme* is a tuple $\kappa = (A, \Sigma, C, P)$ where $C$, the set of *cutting sites*, and $P$, the set of *pasting strings* are subsets of $\Sigma A^* \Sigma$. Given an element of $C$, $c = \alpha u \beta$, and an end-marked string $z = \gamma x u y \delta$, the strings $\gamma x \alpha$ and $\beta y \delta$ are the result of *cutting* $z$ *at the cutting site* $c$. Given an element of $P$, $p = \alpha u \beta$, and two end-marked strings $u = \gamma x \alpha$ and $v = \beta y \delta$ then the string $\gamma x u y \delta$ is the result of *pasting* $u$ *and* $v$ *with the pasting string* $p$. Given an end-marked language $L$, we denote by $\kappa(L)$ the union of $L$ and the set of all end-marked strings that can be obtained by pasting two end-marked strings in $L$ using a pasting string in $P$ or by cutting an end-marked string in $L$ at a cutting site in $C$. We write $\kappa^0(L) = L$. For each integer $i > 0$ we define $\kappa^i(L) = \kappa(\kappa^{i-1}(L))$ and let $\kappa^*(L) = \bigcup_{i=0}^{i=\infty} \kappa^i(L)$. A *CPH-system* is a pair $(M, \kappa)$ where $\kappa$ is a CPH-scheme and $M$ is a set of end-marked strings called the *initial molecules* of the CPH-system. Given an ordered pair of end-markers $(\alpha, \beta) \in \Sigma \times \Sigma$, the $(\alpha, \beta)$*-result* of the system is $\mathrm{Strip}(\kappa^*(M) \cap \alpha A^* \beta)$. $\mathrm{Strip}(\kappa^*(M))$ is called the *full result* of the system. We will also call the $(\alpha, \beta)$-result of a system a *component* of the full result or a *component result*, in keeping with the matrix form for an end-marked language.

An *n test-tube cut, paste and end-marker filtered H-scheme (nCPFH or simply CPFH-scheme)* is a tuple $\tau = (A, \Sigma, T_1, \ldots, T_n)$ where each $T_i$ is a called a *test-tube*, and each $T_i = (\kappa_i, F_i)$ where $\kappa_i = (A, \Sigma, C_i, P_i)$ is a CPH-scheme and $F_i \subset \Sigma \times \Sigma$. We call $F_i$ the *filter* for tube $T_i$. A filter $F \subset \Sigma \times \Sigma$ determines an end-marked language $L(F) = \bigcup_{(\alpha,\beta) \in F} \alpha A^* \beta$. Given an $n$-tuple of end-marked languages $\boldsymbol{L} = (L_1, L_2, \ldots, L_n)$ we define the action of $\tau$ on $\boldsymbol{L}$ by component as

$$(\tau(\boldsymbol{L}))_i = \kappa_i(L_i) \cup \bigcup_{j=1}^{j=n} (\kappa_j(L_j) \cap L(F_i)).$$

Note that $L_i \subset \kappa_i(L_i) \subset (\tau(\boldsymbol{L}))_i$. As before, we define $\tau^0(\boldsymbol{L}) = \boldsymbol{L}$ and, for $i > 0$, $\tau^i(\boldsymbol{L}) = \tau(\tau^{i-1}(\boldsymbol{L}))$. Then we define $\tau^*(\boldsymbol{L}) = \bigcup_{i=0}^{i=\infty} \tau^i(\boldsymbol{L})$. A *CPFH-system* is an $n$-tuple of sets of molecules (or axioms, the initial tube contents) $\boldsymbol{M} = (M_1, \ldots, M_n)$ together with a CPFH-scheme $\tau$. Given an ordered pair of end-markers $(\alpha, \beta) \in \Sigma \times \Sigma$, the $(\alpha, \beta)$*-result* of the system is $\mathrm{Strip}((\tau^*(\boldsymbol{M}))_1 \cap \alpha A^* \beta)$. $\mathrm{Strip}(\tau^*(\boldsymbol{M}))_1$ is called the *full result* of the system. As with CPH-systems, we will call $(\alpha, \beta)$-results *component results.* For multi-test-tube systems, the distinction between full and component results is less significant than it is for single test-tube systems. In fact, given an $(\alpha, \beta)$-result $L$ of a CPFH-system $\tau$, there exists another CPFH-system $\tau'$ with one extra tube (with no molecules, no cutting or pasting, and filter $(\alpha, \beta)$) whose full result is $L$.

We will examine *finite CPFH-systems*, i.e. those in which the various sets, $M_i$ of molecules, $C_i$ of cutting sites, and $P_i$ of pasting strings are all finite. We will study the generative power of the unrestricted finite CPFH-system and special cases where one or more of the facets (cutting, pasting or filtering) act trivially. A CPFH-system *in which there is no filtering* is one in which there is only one test-tube ($n = 1$) and $F = \Sigma \times \Sigma$. A CPFH-system *in which there is no cutting* is one in which all the $C_i$ are empty. A CPFH-system *in which there is no pasting* is one in which all the $P_i$ are empty. Of course, a CPFH-system in which there is no filtering is a CPH-system as defined above. We will analogously refer to CPFH-systems in which there is neither filtering or pasting as *CH-systems*, CPFH-systems in which there is no cutting as *PFH-systems*, etc.

## 2. Some Examples

We present two examples that illustrate the power of filtering, a simple one with pasting but not cutting and a second one involving both cutting and pasting that illustrates some of the techniques used in the construction in Section 4.

The first example is a PFH-scheme with two tubes: $\tau = (A, \Sigma, T_1, T_2)$ with, for $i = 1, 2$, $T_i = (\kappa_i, F_i)$ and $\kappa_i = (A, \Sigma, \emptyset, P_i)$. It has a two letter alphabet $A = \{b, d\}$ and uses three end-markers, $\Sigma = \{\epsilon, \delta, \gamma\}$. The pasting rules are $P_1 = \{\delta\epsilon, \epsilon\delta\}$ and $P_2 = \{\delta\gamma, \gamma\delta\}$. The two filters are $F_1 = \{(\epsilon, \epsilon)\}$ and $F_2 = \{(\gamma, \gamma)\}$. The initial contents are $M_1 = \{\gamma\delta, \delta\gamma\}$ and $M_2 = \{\gamma\gamma, \epsilon b\delta, \delta d\epsilon\}$. First, in test-tube 2, $\epsilon b\delta$ and $\gamma\gamma$ are pasted using $\delta\gamma$ creating $x = \epsilon b\gamma$ and $\gamma\gamma$ and $\delta d\epsilon$ are pasted using $\gamma\delta$ creating $y = \gamma d\epsilon$. Then, either using $x$ and $\delta d\epsilon$ or using $\epsilon b\delta$ and $y$, further pasting creates $\epsilon bd\epsilon$ which passes the filter to the first tube. In test-tube 1 pasting occurs (changing the end-markers) to create $\gamma bd\epsilon$, $\epsilon bd\gamma$ and $\gamma bd\gamma$. The last of these passes the filter for test-tube 2. The cycle can repeat giving the eventual components of $\tau^*(M_1, M_2)$ as follows. The second test-tube contains

$$\{\gamma b^n d^n \gamma, \gamma b^n d^{n+1} \epsilon, \epsilon b^{n+1} d^n \gamma, \epsilon b^{n+1} d^{n+1} \epsilon\}_{n \geq 0} \cup \{\epsilon b\delta, \delta d\epsilon\}.$$

The first test-tube contains

$$\{\epsilon b^n d^n \epsilon, \epsilon b^n d^n \gamma, \gamma b^n d^n \epsilon, \gamma b^n d^n \gamma\}_{n \geq 1} \cup \{\epsilon\delta, \delta\epsilon\}.$$

Note that the non-regular language $\{b^n d^n\}_{n \geq 0}$ is the full result of the system.

The second example is a CPFH-scheme with two tubes: $\tau = (A, \Sigma, T_1, T_2)$ with $T_i = (\kappa_i, F_i)$ and $\kappa_i = (A, \Sigma, C_i, P_i)$ for $i = 1, 2$. It has alphabet $A = \{a, \overline{a}, b, c, d, e, \overline{e}\}$ and uses end-markers $\Sigma = \{\alpha, \beta, \gamma, \delta, \epsilon, \iota, \phi, \rho, \sigma, \tau\}$. Cutting occurs only in the first tube: $C_1 = \{\alpha a\gamma, \gamma de\alpha, \tau ce\alpha, \alpha ad\tau, \alpha\overline{a}d\rho, \rho\overline{e}\alpha, \alpha\overline{a}b\beta, \beta d\overline{e}\alpha\}$ and $C_2 = \emptyset$. For pasting: $P_1 = \{\delta\epsilon, \epsilon\delta, \phi\beta, \beta\phi\}$ and $P_2 = \{\delta\gamma, \gamma\delta, \sigma\tau, \tau\sigma, \iota\rho, \rho\iota\}$. The filters are $F_1 = \{(\alpha, \alpha)\}$ and $F_2 = \{(\gamma, \gamma), (\tau, \tau), (\rho, \rho)\}$ The initial contents are $M_1 = \{\epsilon\epsilon, \epsilon bcd\epsilon, \alpha ab\delta, \delta de\alpha, \epsilon b\phi, \phi d\epsilon\}$, $M_2 = \{\alpha ad\delta, \delta e\alpha, \alpha\overline{a}d\sigma, \sigma cc\overline{e}\alpha, \alpha\overline{a}\iota, \iota d\overline{e}\alpha\}$. The $(\epsilon, \epsilon)$-result of this system is $\{b^n c^n d^n\}_{n \geq 0}$. Starting with $\epsilon b^n c^n d^n \epsilon$ in the first tube, we can generate $\epsilon b^{n+1} c^{n+1} d^{n+1} \epsilon$ as follows. Pasting in the first test-tube creates $\alpha ab^{n+1} c^n d^{n+1} e\alpha$. We now use the cutting sites $\alpha a\gamma$ and $\gamma de\alpha$ in the first test-tube to produce $\gamma b^{n+1} c^n d^n \gamma$ which is filtered into the second tube where pasting occurs to produce $\alpha adb^{n+1} c^n d^n e\alpha$ which is filtered back into the first tube. This cut, filter, paste, filter sequence has rotated one of the $d$'s from the end of the word $b^{n+1} c^n d^{n+1}$ to the beginning. This sort of rotation has been used by Păun and others [1, 7, 10, 3] in the proofs that various spicing and cutting and recombination systems can generate any RE language. We will use it in section 4 ourselves. We can repeat the sequence $n$ times to eventually obtain $\alpha ad^{n+1} b^{n+1} c^n e\alpha$. This is cut at the two cutting sites $\tau ce\alpha$ and $\alpha ad\tau$ to produce $\tau d^n b^{n+1} c^{n-1} \tau$ which is filtered to the second tube where pasting creates $\alpha\overline{a}d^{n+1} b^{n+1} c^{n+1} \overline{e}\alpha$ which is filtered back into the first test-tube. The cutting sites $\alpha\overline{a}d\rho$ and $\rho\overline{e}\alpha$ work together with filtering and the unused axioms and pasting rules in the second tube to rotate the $d$'s back to the end of the word eventually giving $\alpha\overline{a}b^{n+1}c^{n+1}d^{n+1}\overline{e}\alpha$ in the first tube. Cutting this at the final two cutting sites in the first tube produces $\beta b^n c^{n+1} d^n \beta$, which can be pasted together with the unused axioms $\epsilon b\phi$ and $\phi d\epsilon$ to produce $\epsilon b^{n+1}c^{n+1}d^{n+1}\epsilon$.

## 3. Summary of Generative Capacity Results

**Proposition 3.1.** *Each component result of a finite CPFH-system is recursively enumerable and any recursively enumerable set can be obtained as the full result of a finite CPFH-system with three test-tubes or as a component result of a finite CPFH-system with two test-tubes.*

*Proof.* The proof is found in Section 4. ☐

**Proposition 3.2.** *Each component result of a finite PFH-system is context free and any context free set can be obtained as the full result of a finite PFH-system.*

*Proof.* The proof is found in Section 5. ☐

**Proposition 3.3.** *Each component result of a finite CPH-system (and therefore a finite PH-system) is a regular set, and any regular set can be obtained as one of the component results of a finite PH-system (and therefore as one of the component results of a finite CPH-system).*

*Proof.* This is established in Pixton's work [9]. The first is part of his Closure Theorem and the construction he gives (in his proposition 7.1) to show that any regular set can be obtained as the result of a CPH-scheme uses a scheme in which there is no cutting. ☐

**Proposition 3.4.** *Each component result of a finite CFH-system (and therefore a finite CH-system or a finite FH-system) is finite, and any finite end-marked language can be obtained as the (unstripped) result of a finite FH-system with only one test-tube (and therefore as the result of a finite CH-system or CFH-system).*

*Proof.* Let $\boldsymbol{M} = (M_1, \dots, M_n)$ and $\tau = (A, \Sigma, T_1, \dots, T_n)$ comprise a finite CFH-system, with $T_i = (\kappa_i, F_i)$ and $\kappa_i = (A, \Sigma, C_i, \emptyset)$. Since there is no pasting, the result of the system is a subset of $\Sigma(\mathrm{Fac}(\bigcup_{i=1}^{i=n} \mathrm{Strip}(M_i)))\Sigma$, a finite set.

If $S \subset \Sigma A^* \Sigma$ is a finite set, then a trivial finite FH-system that has $S$ as a result is the one-test-tube system with $M_1 = S$, $F_1 = \Sigma \times \Sigma$ and $\kappa_1 = (A, \Sigma, \emptyset, \emptyset)$. ☐

## 4. CPFH-systems generate exactly the RE Languages

In this section we prove Proposition 3.1. The proof follows the ideas of Csuhaj-Varjú, Kari, and Păun [1], and Priese, Rogojine and Margenstern [10] who worked with multi-test-tube splicing systems and determined that such systems could generate any RE language. The proof also uses the ideas of Freund and Freund [3] who recently showed that multi-test-tube cutting and recombination systems could generate any RE language in just two test-tubes with a final filtering step. We establish similar results for CPFH-systems.

Let $G = (N, T, R, S)$ be a type-0 grammar. Find new symbols $B, l, r \notin N \cup T$ and let $A = N \cup T \cup \{B, l, r\}$ and $A' = N \cup T \cup \{B\}$. Create a set of distinct new symbols (to be used as end-markers) $\Sigma = \{\nabla, \Delta, \epsilon, \delta\} \cup \{\delta_u \mid u \to v \in R\} \cup \{\gamma_a, \sigma_a \mid a \in A'\}$ with $\Sigma \cap A = \emptyset$.

We first define a CPFH-system with 3 test-tubes, with initial molecules $\boldsymbol{M} = (M_1, M_2, M_3)$ and a CPFH-scheme $\tau = (A, \Sigma, T_1, T_2, T_3)$ with $T_i = (\kappa_i, F_i)$ and $\kappa_i = (A, \Sigma, C_i, P_i)$. This

CPFH-scheme will produce in test-tube 1 all the words $\nabla w \nabla$ where $w \in (N \cup T)^*$ such that $S \stackrel{*}{\to} w$. The system uses an idea of Păun, rotating words so that we can always derive (using cutting and pasting) only on the right side of a sentential form. This enables us to retain control of the derivations. Later we will modify this system so that only $\nabla L(G) \nabla$ is in the first test-tube.

To define the CPFH-system we must specify (for each $i$) $M_i$, $F_i$, $C_i$ and $P_i$. For test-tube 1 (used to collect final results) let

$$M_1 = \emptyset, F_1 = \{(\nabla, \nabla)\}, C_1 = \emptyset, \text{and} P_1 = \emptyset.$$

For test-tube 2 (used for deriving the sentential forms and finishing the rotations of symbols in $A'$) let

$$\begin{aligned} &M_2 = \{\Delta l B S r \Delta, \epsilon\Delta, \Delta\epsilon\}, F_2 = \{(\gamma_a, \sigma_a) \mid a \in A'\}, C_2 = \{\delta_u u r \epsilon \mid u \to v \in R\}, \\ &\text{and } P_2 = \{\delta_u v r \epsilon \mid u \to v \in R\} \cup \{\epsilon l a \gamma_a, \sigma_a r \epsilon \mid a \in A'\}. \end{aligned}$$

Finally for test-tube 3 (used to begin the rotations of symbols in $A'$ and to prepare entries for filtering into the first test-tube) let

$$M_3 = P_3 = \emptyset, F_3 = \{(\Delta, \Delta)\}, \text{ and } C_3 = \{\sigma_a a r \epsilon, \epsilon l \gamma_a \mid a \in A'\} \cup \{\epsilon l B \nabla, \nabla r \epsilon\}.$$

**Claim.** $\tau^*(\boldsymbol{M}) = \boldsymbol{D}$ where

$$\begin{aligned} D_1 &= \{\nabla w \nabla \mid S \stackrel{*}{\to} w\}, \\ D_2 &= \{\Delta l w_2 B w_1 r \Delta \mid S \stackrel{*}{\to} w_1 w_2\} \cup \{\Delta l w_2 B w_1 \delta_u \mid S \stackrel{*}{\to} w_1 u w_2, u \to v \in R\} \\ &\quad \cup \{\gamma_a w_2 B w_1 \sigma_a, \Delta l a w_2 B w_1 \sigma_a, \gamma_a w_2 B w_1 r \Delta, \Delta l a w_2 B w_1 r \Delta, \gamma_a w_2 B w_3 \delta_u \\ &\qquad\quad \mid S \stackrel{*}{\to} w_1 a w_2, u \to v \in R, w_1 = w_3 u\} \\ &\quad \cup \{\gamma_B w \sigma_B, \Delta l B w \sigma_B, \gamma_B w r \Delta, \gamma_B w_1 \delta_u \mid S \stackrel{*}{\to} w, u \to v \in R, w = w_1 u\} \\ &\quad \cup \{\epsilon\Delta, \Delta\epsilon\}, \\ D_3 &= \{\Delta l w_2 B w_1 r \Delta, \gamma_b w_2 B w_1 r \Delta, \Delta l w_2 B w_1 \nabla, \gamma_b w_2 B w_1 \nabla \mid S \stackrel{*}{\to} w_1 w_2, b \in A'\} \\ &\quad \cup \{\Delta l w_2 B w_1 \sigma_a, \gamma_b w_2 B w_1 \sigma_a \mid S \stackrel{*}{\to} w_1 a w_2, b \in A'\} \\ &\quad \cup \{\gamma_b w B r \Delta, \gamma_b w \sigma_B, \Delta l w \sigma_B, \nabla w r \Delta, \nabla w \nabla \mid S \stackrel{*}{\to} w, b \in A'\} \cup \{\epsilon\Delta, \Delta\epsilon\}. \end{aligned}$$

*Proof of Claim.* Note that $\boldsymbol{M} \subset \boldsymbol{D}$. To see $\boldsymbol{D} \subset \tau^*(\boldsymbol{M})$, we first establish three lemmas which describe the action of $\tau$.

**Lemma 4.1.** *(Final Filtering) Let* $\tau$ *be as above,* $x \in (N \cup T)^*$*,* $\boldsymbol{E}$ *a tuple of end-marked languages (test-tube contents for* $\tau$*) and let* $\Delta l B x r \Delta \in E_2$*. Then* $\Delta l B x r \Delta$*,* $\nabla x r \Delta$*,* $\Delta l B x \nabla$ *and* $\nabla x \nabla$ *are in* $(\tau^*(\boldsymbol{E}))_3$ *and* $\nabla x \nabla$ *is in* $(\tau^*(\boldsymbol{E}))_1$

*Proof.* Since $y = \Delta l B x r \Delta \in E_2 \subset (\tau(\boldsymbol{E}))_2$ and $y \in L(F_3)$ we have $y$ filtered into $(\tau(\boldsymbol{E}))_3$. Cutting $y$ in test-tube 3 at sites $\epsilon l B \nabla$ and $\nabla r \epsilon$ produces $\nabla x r \Delta$ and $\Delta l B x \nabla$ in $(\tau^2(\boldsymbol{E}))_3$ and $\nabla x \nabla \in (\tau^3(\boldsymbol{E}))_3$. Then, filtering, we have $\nabla x \nabla \in (\tau^3(\boldsymbol{E}))_1$. ☐

**Lemma 4.2.** *(Rotation) Let* $\tau$ *be as above,* $a \in A'$*,* $xa \in A^*$*,* $\boldsymbol{E}$ *a tuple of end-marked languages and let* $\Delta l x a r \Delta \in E_2$*. Then* $\Delta l x a r \Delta, \gamma_b x a r \Delta$*,* $\Delta l x \sigma_a$*, and* $\gamma_b x \sigma_a$ *(for each* $b \in A'$*) are all in* $(\tau^*(\boldsymbol{E}))_3$ *and* $\gamma_a x \sigma_a$*,* $\gamma_a x r \Delta$*,* $\Delta l a x \sigma_a$*, and* $\Delta l a x r \Delta$ *are in* $(\tau^*(\boldsymbol{E}))_2$*. More generally,* $\Delta l x_1 x_2 r \Delta \in E_2$ *implies* $\Delta l x_2 x_1 r \Delta \in (\tau^*(\boldsymbol{E}))_2$*.*

*Proof.* Start with $y = \Delta lxar\Delta \in E_2$ which is filtered into test-tube 3. Cut $y$ in test-tube 3 at the sites $\epsilon l\gamma_b$ and $\sigma_a ar\epsilon$ to produce $\gamma_b xar\Delta$ and $\Delta lx\sigma_a$ in $(\tau(\boldsymbol{E}))_3$ and $\gamma_b x\sigma_a$ in $(\tau^2(\boldsymbol{E}))_3$. (Note that many $\gamma_b x\sigma_a$'s are produced – those with $a \neq b$ are garbage.) With filtering, only $\gamma_a x\sigma_a$ passes into $(\tau^2(\boldsymbol{E}))_2$. Using the pasting rules in $P_2$ $\epsilon la\gamma_a$ and $\sigma_a r\epsilon$ on the contents of $M_2$ and $\gamma_a x\sigma_a$ produces both $\gamma_a xr\Delta$ and $\Delta lax\sigma_a$ in $(\tau^3(\boldsymbol{E}))_2$ and $\Delta laxr\Delta$ in $(\tau^4(\boldsymbol{E}))_2$. To establish the last statement of the lemma, rotate a total of $|x_2|$ letters, putting $\Delta lx_2x_1r\Delta \in (\tau^{4|w_2|}(\boldsymbol{E}))_2$. ☐

**Lemma 4.3.** *(Derivation) Let $\tau$ be as above, $u \to v \in R$, $\boldsymbol{E}$ a tuple of end-marked languages and let $\Delta lxur\Delta \in E_2$. Then $\Delta lx\delta_u$ and $\Delta lxvr\Delta \in (\tau^*(\boldsymbol{E}))_2$.*

*Proof.* Consider $y = \Delta lxur\Delta \in E_2$. Cutting the string $y$ in test-tube 2 at cutting site $\delta_u ur\epsilon$ produces $z = \Delta lx\delta_u \in (\tau(\boldsymbol{E}))_2$. Pasting $z$ and $\epsilon\Delta$ with $\delta_u vr\epsilon$ produces $\Delta lxvr\Delta \in (\tau^3(\boldsymbol{E}))_2$. ☐

Now we have a fourth lemma which we will prove inductively.

**Lemma 4.4.** *If $w \in (N \cup T)^*$ with $S \xrightarrow{*} w$ then $\Delta lBwr\Delta \in (\tau^*(\boldsymbol{M}))_2$*

*Proof.* We proceed by induction on the number of steps used in deriving $w$. As a base, consider $w = S$. Since $\Delta lBSr\Delta \in M_2$ we have our result. Now, for the step, assume our conclusion holds for all sentential forms in $(N \cup T)^*$ which can be derived from $S$ in fewer than $k$ steps and suppose $S \xrightarrow{k} w$ where the last production used is $u \to v$, i.e. $S \xrightarrow{k-1} w_1uw_2 \to w_1vw_2 = w$. Then our inductive hypothesis, applied to $w_1uw_2$, gives $\Delta lBw_1uw_2r\Delta \in (\tau^*(\boldsymbol{M}))_2$. Rotating $w_2$ with Lemma 4.2 gives $\Delta lw_2Bw_1ur\Delta \in (\tau^*(\boldsymbol{M}))_2$. Then, using Lemma 4.3 we get $\Delta lw_2Bw_1vr\Delta \in (\tau^*(\boldsymbol{M}))_2$. Finally rotating $Bw_1v$ with Lemma 4.2 gives $\Delta lBw_1vw_2r\Delta = \Delta lBwr\Delta$ in test-tube 2. ☐

To see that $D_1 \subset (\tau^*(\boldsymbol{M}))_1$, suppose $w \in (N \cup T)^*$ such that $S \xrightarrow{*} w$. Lemma 4.4 gives $\Delta lBwr\Delta \in (\tau^*(\boldsymbol{M}))_2$ and then Lemma 4.1 gives $\nabla w\nabla \in (\tau^*(\boldsymbol{M}))_1$.

To see that $D_2 \subset (\tau^*(\boldsymbol{M}))_2$ consider the contents of $D_2$. First, if $S \xrightarrow{*} w_1w_2$ then by Lemma 4.4 we have $\Delta lBw_1w_2r\Delta$ in $(\tau^*(\boldsymbol{M}))_2$ and rotating $w_2$ with Lemma 4.2 gives $\Delta lw_2Bw_1r\Delta$ in $(\tau^*(\boldsymbol{M}))_2$ as we want. Next suppose $S \xrightarrow{*} w_1uw_2$ and $u \to v \in R$. Then $\Delta lBw_1uw_2\Delta$ in $(\tau^*(\boldsymbol{M}))_2$ and with rotation $\Delta lw_2Bw_1u\Delta$ is in $(\tau^*(\boldsymbol{M}))_2$. Using Lemma 4.3 gives $\Delta lw_2Bw_1\delta_u \in (\tau^*(\boldsymbol{M}))_2$ as desired. Now suppose $S \xrightarrow{*} w_1aw_2$ with $u \to v \in R$ and $w_1 = w_3u$. Then, again using Lemma 4.4 and Lemma 4.2, we have $\Delta lw_2Bw_1ar\Delta$ in $(\tau^*(\boldsymbol{M}))_2$. Using Lemma 4.2 again with $x = w_2Bw_1$ we get $\gamma_a w_2Bw_1\sigma_a$, $\gamma_a law_2Bw_1r\Delta$, $\Delta law_2Bw_1\sigma_a$ and $\Delta law_2Bw_1r\Delta$ in $(\tau^*(\boldsymbol{M}))_2$. Applying Lemma 4.3 to $\gamma_a w_2Bw_1r\Delta = \gamma_a w_2Bw_3ur\Delta$ shows $\gamma_a w_2Bw_3\delta_u$ is in $(\tau^*(\boldsymbol{M}))_2$. Finally, if $S \xrightarrow{*} w$ with $w = w_1u$ and $u \to v \in R$, using Lemma 4.4 and then rotating $w$ using Lemma 4.2 we have $\Delta lwBr\Delta$ in $(\tau^*(\boldsymbol{M}))_2$. Applying Lemma 4.2 again with $w$ for the $x$ in the lemma and $B$ for the $a$ in the lemma we see $\gamma_B w\sigma_B$, $\gamma_B wr\Delta$ and $\Delta lBw\sigma_B$ are in $(\tau^*(\boldsymbol{M}))_2$. Then using Lemma 4.3 to $\gamma_B wr\Delta = \gamma_B w_1ur\Delta$ gives us $\gamma_B w_1\delta_u \in (\tau^*(\boldsymbol{M}))_2$. Lastly, note that $\epsilon\Delta$ and $\Delta\epsilon$ are part of the initial contents of test-tube 2.

To see that $D_3 \subset (\tau^*(\boldsymbol{M}))_3$ consider the contents of $D_3$. First suppose $S \xrightarrow{*} w_1w_2$ and $b \in A'$. Lemma 4.4 and Lemma 4.2 give us $\Delta lw_2Bw_1r\Delta$ in $(\tau^*(\boldsymbol{M}))_2$. Filtering gives $\Delta lw_2Bw_1r\Delta \in (\tau^*(\boldsymbol{M}))_3$ as well where it can be cut either at site $\epsilon l\gamma_b$ or $\nabla r\epsilon$ or both to produce $\gamma_b w_2 B w_1 r\Delta$, $\Delta l w_2 B w_1 \nabla$ and $\gamma_b w_2 B w_1 \nabla$ in $(\tau^*(\boldsymbol{M}))_3$ as desired. Next suppose $S \xrightarrow{*} w_1 a w_2$ and $b \in A'$. Once again our lemmas give $\Delta l w_2 B w_1 a r \Delta$ in $(\tau^*(\boldsymbol{M}))_2$ and filtering gives $\Delta l w_2 B w_1 a r \Delta$ in $(\tau^*(\boldsymbol{M}))_3$ as well. Cutting at site $\sigma_a a r \epsilon$ and then at site $\epsilon l \gamma_b$ gives $\Delta l w_2 B w_1 \sigma_a$ and $\gamma_b w_2 B w_1 \sigma_a$ in $(\tau^*(\boldsymbol{M}))_3$. Finally, if $S \xrightarrow{*} w$ and $b \in A'$, Lemma 4.4 gives $\Delta l B w r \Delta$ in $(\tau^*(\boldsymbol{M}))_2$. Applying Lemma 4.1 gives $\Delta l B w r \Delta$, $\Delta l B w \nabla$ and $\nabla w \nabla$ in $(\tau^*(\boldsymbol{M}))_3$. Cutting $\Delta l B w r \Delta$ at cutting site $\epsilon l \gamma_b$ or site $\sigma_B B r \epsilon$ or both gives $\gamma_b w B r \Delta$, $\Delta l w \sigma_B$ and $\gamma_b w \sigma_B$ in $(\tau^*(\boldsymbol{M}))_3$. Lastly, note that $\epsilon\Delta$ and $\Delta\epsilon$ are produced by cutting $\Delta l B S r \Delta$ at the sites $\epsilon l B \nabla$ and $\nabla r \epsilon$.

To finish the proof of the claim, we observe that $\tau(\boldsymbol{D}) \subset \boldsymbol{D}$. By inspection, for each $i, j \in \{1,2,3\}$, $\kappa_i(D_i) \subset D_i$ and $D_i \cap L(F_j) \subset D_j$. ☐

We now modify this CPFH-system to get a new CPFH-system $\hat{\tau}$ that has only $\nabla L(G) \nabla$ as the final contents of test-tube 1. We will filter the sentential forms keeping only those in $\nabla T^* \nabla$. Most of $\hat{\tau}$ is the same as $\tau$ above, but we modify test-tubes 2 and 3 to enable us to filter out sentential forms not in $T^*$. We need one new symbol, $\hat{l}$, and new end-markers: $\hat{\Delta}$, $\hat{\epsilon}$ and a marker $\hat{\gamma}_a$ for each $a \in T$. Let $\hat{M}_2 = M_2 \cup \{\hat{\Delta}\hat{\epsilon}\}$, $\hat{C}_2 = C_2$, $\hat{F}_2 = F_2 \cup \{(\hat{\gamma}_a, \sigma_a) \mid a \in T\}$ and $\hat{P}_2 = P_2 \cup \{\hat{\epsilon}\hat{l}a\hat{\gamma}_a \mid a \in T\}$. Also let $\hat{M}_3 = \{\hat{\Delta}\delta\}$,$\hat{F}_3 = F_3 \cup \{(\hat{\Delta}, \Delta)\}$, $\hat{P}_3 = \{\delta\hat{l}B\delta\}$ and let

$$\hat{C}_3 = C_3 \cup \{\epsilon l B \delta, \epsilon\hat{l}\nabla, \nabla B r \epsilon\} \cup \{\epsilon\hat{l}\hat{\gamma}_a \mid a \in T\} - \{\epsilon l B \nabla, \nabla r \epsilon\}$$

Then the new tube contents are:

$$\begin{aligned}
\hat{D}_1 \;&=\; \{\nabla w \nabla \mid S \xrightarrow{*} w, w \in T^*\}, \\
\hat{D}_2 \;&=\; \{\Delta l w_2 B w_1 r \Delta \mid S \xrightarrow{*} w_1 w_2\} \cup \{\Delta l w_2 B w_1 \delta_u \mid S \xrightarrow{*} w_1 u w_2, u \rightarrow v \in R\} \\
&\quad \cup\{\gamma_a w_2 B w_1 \sigma_a, \Delta l a w_2 B w_1 \sigma_a, \gamma_a w_2 B w_1 r \Delta, \Delta l a w_2 B w_1 r \Delta, \gamma_a w_2 B w_3 \delta_u \\
&\qquad \mid S \xrightarrow{*} w_1 a w_2, u \rightarrow v \in R, w_1 = w_3 u\} \\
&\quad \cup\{\hat{\gamma}_a w_2 B w_1 \sigma_a, \hat{\Delta}\hat{l} a w_2 B w_1 \sigma_a, \hat{\gamma}_a w_2 B w_1 r \Delta, \hat{\Delta}\hat{l} a w_2 B w_1 r \Delta, \hat{\gamma}_a w_2 B w \delta_u \\
&\qquad \mid S \xrightarrow{*} w_1 a w_2, a \in T, w_2 \in T^*, u \rightarrow v \in R, w_1 = w_3 u\} \\
&\quad \cup\{\gamma_B w \sigma_a, \Delta l B w \hat{\sigma}_B, \gamma_B w r \Delta, \gamma_B w_1 \delta_u \mid S \xrightarrow{*} w, u \rightarrow v \in R, w = w_1 u\} \\
&\quad \cup\{\epsilon\Delta, \Delta\epsilon, \hat{\Delta}\hat{\epsilon}\}, \text{ and}
\end{aligned}$$

$$\begin{aligned}
\hat{D}_3 \;&=\; \{\Delta l w_2 B w_1 r \Delta, \gamma_b w_2 B w_1 r \Delta \mid S \xrightarrow{*} w_1 w_2, b \in A'\} \\
&\quad \cup\{\Delta l w_2 B w_1 \sigma_a, \gamma_b w_2 B w_1 \sigma_a \mid S \xrightarrow{*} w_1 a w_2, b \in A'\} \\
&\quad \cup\{\Delta l w \nabla, \gamma_b w B r \Delta, \gamma_b w \nabla, \gamma_b w r \sigma_B, \delta w r \Delta, \delta w \sigma_B, \hat{\Delta}\hat{l} B w r \Delta, \hat{\Delta}\hat{l} w \sigma_B \\
&\qquad \mid S \xrightarrow{*} w, b \in A'\} \\
&\quad \cup\{\hat{\Delta}\hat{l} w_2 B w_1 r \Delta, \nabla w_2 B w_1 r \Delta, \hat{\gamma}_b w_2 B w_1 r \Delta \mid S \xrightarrow{*} w_1 w_2, w_2 \in T^*, b \in T\} \\
&\quad \cup\{\hat{\Delta}\hat{l} w_2 B w_1 \sigma_a, \hat{\gamma}_b w_2 B w_1 \sigma_a, \nabla w_2 B w_1 \sigma_a \\
&\qquad \mid S \xrightarrow{*} w_1 a w_2, a \in T, b \in T, w_2 \in T^*\} \\
&\quad \cup\{\hat{\gamma}_b w \nabla, \hat{\Delta}\hat{l} w \nabla, \nabla w \nabla \mid S \xrightarrow{*} w, w \in T^*, b \in T\} \cup \{\epsilon\Delta, \Delta\epsilon, \hat{\Delta}\delta\}.
\end{aligned}$$

Test-tube 3 collects, as before, all strings of the form $\Delta l w_2 B w_1 r \Delta$ such that $S \xrightarrow{*} w_1 w_2$, $w_1 w_2 \in (N \cup T)^*$. Cutting and pasting in test-tube 3 replaces the $\Delta l$ on the left with $\hat{\Delta}\hat{l}$ only on those strings that have the $B$ at the left of the sentential form. The strings created by this cutting and pasting (those of the form $\hat{\Delta}\hat{l}Bwr\Delta$ where $S \xrightarrow{*} w$, $w \in (N \cup T)^*$) are rotated using the additions to test-tube 3 and test-tube 2 involving the new end-markers $\hat{\gamma}_a$ (for $a$ in $T$) and the new symbol $\hat{l}$. Only those sentential forms $w$ which are in $T^*$ will be completely rotated to $\hat{\Delta}\hat{l}wBr\Delta$. These completely rotated forms are then re-end-marked and filtered into test-tube 1 in the same way that the final results of $\tau$ were filtered into test-tube 1. A proof similar to the one for $\tau$ above can be used to establish these tube contents. This finishes the proof of Proposition 3.1.

Note that in both $\tau$ and $\hat{\tau}$ test-tube 1 is used just to collect the final results. Alternatively, the first tube could be discarded and the remaining tubes renumbered so that the desired result would be the $(\nabla, \nabla)$-result of either $\tau$ or $\hat{\tau}$.

## 5. PFH-systems generate exactly the CF Languages

In this section we prove Proposition 3.2. We will first construct a PFH-system to generate an arbitrary CF-language as its result. Let $L \subset A^*$ be a CF-language and let $G = (N, A, R, S)$ be a grammar for $L$ in a modified Chomsky normal form, in which all the productions have the form $v \to st$ (where $v, s, t$ are distinct non-terminals) or $v \to x$ where $v \in N$ and $x \in A \cup \{1\}$. Such a grammar can be obtained from one in Chomsky normal form quite easily. If a production of the form $q \to rr$ exists, replace it with the productions $q \to rs$, $s \to rt$ and $t \to 1$ where $s$ and $t$ are new non-terminals. Productions of the forms $q \to qr$ and $q \to rq$ can be similarly replaced.

We now define our PFH-system with $n = |R| + 1$ test-tubes, with initial molecules $\boldsymbol{M} = (M_1, \dots, M_n)$ and a PFH-scheme $\tau = (A, \Sigma, T_1, \dots, T_n)$ with $T_i = (\kappa_i, F_i)$ and $\kappa_i = (A, \Sigma, \emptyset, P_i)$ where $\Sigma = \{\epsilon_v, \delta_v \mid v \in N\}$. We will denote all test-tubes but $T_1$ as $T_p$ where $p \in R$.

The first test-tube has filter $F_1 = (\epsilon_S, \delta_S)$ and no pasting rules or initial contents. It will collect the finished words in the CF-language.

For each $p \in R$ with form $v \to x$ where $x \in A \cup \{1\}$, let

$$F_p = P_p = \emptyset \text{ and } M_p = \{\epsilon_v x \delta_v\}.$$

For each $p \in R$ with form $v \to st$ where $v, s, t \in N$ are all distinct, let

$$F_p = \{(\epsilon_s, \delta_s), (\epsilon_t, \delta_t)\}, P_p = \{\epsilon_v\epsilon_s, \delta_s\epsilon_t, \delta_t\delta_v\} \text{ and } M_p = \{\epsilon_v\epsilon_v, \delta_v\delta_v\}.$$

We will show that for each production $p \in R$, with left-hand side $v$, the corresponding tube $T_p$ will eventually produce all the end-marked strings $\epsilon_v w \delta_v$ where $w \in A^*$ such that there exists $w' \in A^*$ with $v \xrightarrow{p} w' \xrightarrow{*} w$. A bit more notation will be useful: for $s \in N$ define $L_s = \{w \in A^* \mid s \xrightarrow{*} w\}$. Then the tube contents can be described as follows.

For $p$ with the form $v \to st$ where $v, s, t$ are three distinct non-terminals, let

$$D_p = \{\epsilon_s, \epsilon_v\}L_s\delta_s \cup \epsilon_t L_t\{\delta_t, \delta_v\} \cup \{\epsilon_s, \epsilon_v\}L_s L_t\{\delta_t, \delta_v\} \cup \{\epsilon_v\epsilon_v, \delta_v\delta_v\}.$$

For $p$ of the form $v \to x$ where $v \in N$ and $x \in A \cup \{1\}$ let $D_p = M_p$.

Finally, let $D_1 = \epsilon_S L \delta_S$.

We will argue that $\boldsymbol{D} = \tau^*(\boldsymbol{M})$.

It is trivial to check that $\boldsymbol{M} \subset \boldsymbol{D}$. Note that no pasting occurs in test-tube 1 or in test-tube $p$ for $p$ of the form $v \to x$. Furthermore, since $v$, $s$ and $t$ are all distinct for $p$ of the form $v \to st$, it is easy to see that $D_p$ is closed under pasting for any such $p$. Finally, for each pair of test-tubes $T_i$, $T_j$, anything in $D_i$ that passes the filter for $T_j$ is already in $D_j$. Thus, $\boldsymbol{D}$ is closed under $\tau$ so that $\tau^*(\boldsymbol{M}) \subset \boldsymbol{D}$.

To see that $\boldsymbol{D} \subset \tau^*(\boldsymbol{M})$ we shall show that if $w \in A^*$, $p \in R$ and $\alpha, \beta \in \Sigma$ with $\alpha w \beta \in D_p$, we have $\alpha w \beta \in (\tau^*(\boldsymbol{M}))_p$ as well. First we have a lemma:

**Lemma 5.1.** *If* $v \in N$ *and* $w \in A^*$ *with* $v \xrightarrow{*} w$ *then* $\epsilon_v w \delta_v$ *is in some component of* $\tau^*(\boldsymbol{M})$, *i.e. there is a* $p \in R$ *with left hand side* $v$ *such that* $\epsilon_v w \delta_v \in \tau^*(\boldsymbol{M})_p$.

*Proof.* If $v \in N$ and $w \in A^*$ with $v \to w$ (in one step), then $w \in A \cup \{1\}$ and $v \to w$ is the needed production $p$. Now let $m > 1$ and suppose our lemma holds for all $v \in N$, $w \in A^*$ where $v \xrightarrow{n}$ with $1 \leq n < m$ and let $v \xrightarrow{m} w$ and let $v \to w_1 \xrightarrow{m-1} w$ be one of the derivations of length $m$. Then, since $v$ is a single non-terminal, $v \to w_1$ must be a production in $R$, and since $m > 1$, the production $v \to w_1$ must be of the form $v \to st$ where $v$, $s$ and $t$ are all distinct.

Since $st \xrightarrow{*} w$ and $L$ is context-free, we must have $w = w_s w_t$ where $w_s \in L_s$ and $w_t \in L_t$ and since $st \xrightarrow{m-1} w$ the derivations $s \xrightarrow{*} w_s$ and $t \xrightarrow{*} w_t$ each require at most $m-1$ steps. Thus our inductive hypothesis gives us two productions, $p_s$ and $p_t$ with $\epsilon_s w_s \delta_s \in \tau^*(\boldsymbol{M})_{p_s}$ and $\epsilon_t w_t \delta_t \in \tau^*(\boldsymbol{M})_{p_t}$. The strings $\epsilon_s w_s \delta_s$ and $\epsilon_t w_t \delta_t$ will be filtered into $\tau^*(\boldsymbol{M})_{v \to st}$ and the pasting rules for $T_{v \to st}$ will assemble $\epsilon_v w_s w_t \delta_v$ from these two strings and initial tube contents. So $v \to st$ is the needed production. $\square$

Now let $w \in A^*$, $p \in R$ and $\alpha, \beta \in \Sigma$ so that $\alpha w \beta \in D_p$. If $p$ is of the form $v \to x$ where $x \in A \cup \{1\}$ then $\alpha w \beta = \epsilon_v x \delta_v$ which is in $M_p$ and therefore in $\tau^*(\boldsymbol{M})_p$. If $p$ is of the form $v \to st$ we consider several cases. In the first case, if $\beta = \delta_s$ then $\alpha$ is either $\epsilon_s$ or $\epsilon_v$. Furthermore, we have $w \in L_s$ and by Lemma 5.1 $\epsilon_s w \delta_s$ is in some component of $\tau^*(\boldsymbol{M})$ and will be filtered into $(\tau^*(\boldsymbol{M}))_p$. If $\alpha = \epsilon_s$ then we have $\alpha w \beta \in (\tau^*(\boldsymbol{M}))_p$ directly and if $\alpha = \epsilon_v$ pasting $\epsilon_v \epsilon_v$ with $\epsilon_s w \delta_s$ will give $\alpha w \beta \in (\tau^*(\boldsymbol{M}))_p$. The second case, $\alpha = \epsilon_t$, is symmetrical to the first case. In the third case, when $\alpha = \delta_v$ or $\beta = \epsilon_v$, we have $w = 1$ and $\alpha w \beta \in M_p \subset \tau^*(\boldsymbol{M})_p$. In the final case, we have $w \in L_s L_t$, so we can write $w = w_s w_t$ and by Lemma 5.1, the strings $\epsilon_s w_s \delta_s$ and $\epsilon_t w_t \delta_t$ each occur in some component of $\tau^*(\boldsymbol{M})$ and both will be filtered into $\tau^*(\boldsymbol{M})_p$. There pasting will create $\alpha w \beta$. Thus, for each $p$, $D_p \subset (\tau^*(\boldsymbol{M}))_p$.

Finally note that filtering will move all the contents of $D_1 = \epsilon_S L \delta_S$ into $(\tau^*(\boldsymbol{M}))_1$ as they are produced. Thus $\boldsymbol{D} \subset \tau^*(\boldsymbol{M})$.

Thus we can generate any CF-language as the result of a PFH-system.

To see that any result of a PFH-system must be context free, we will show that every component of the contents of every tube is context free. It will be convenient to use matrix representations of the end-marked languages in the tubes. We can represent end-marked languages in matrix form as follows. Enumerate $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$. For each $L \subset \Sigma A^* \Sigma$ there is an $m \times m$ matrix $M_L$, whose $(i,j)$ component is the subset of $A^*$ satisfying $\sigma_i (M_L)_{ij} \sigma_j = L \cap \sigma_i A^* \sigma_j$. Addition of these matrices corresponds to union of end-marked languages and matrix multiplication corresponds to a selective concatenation operation in which stings are concatenated if they have matching end-markers as follows: Given two end-marked $L, P \subset \Sigma A^* \Sigma$ then $\sigma_i w \sigma_j$ will be in the end-marked language corresponding to $M_L M_P$ if and only if we can write $w = w_1 w_2$ so that there exists a $\sigma_k \in \Sigma$ with $\sigma_i w_1 \sigma_k \in L$

and $\sigma_k w_2 \sigma_j \in P$. Given a family of languages $\mathcal{F}$ we say an end-marked language $L$ is in $\mathcal{F}$ provided that each of the corresponding matrix components $(M_L)_{ij}$ is in $\mathcal{F}$. Pixton describes how to represent a CPH-system in matrix form [9]. We will follow his work, adding a method of representing filtering, but skipping his representation of cutting since we are presently concerned only with pasting and filtering.

Let $\tau$ be a PFH-scheme with $n$ tubes: $\tau = (A, \Sigma, T_1, \ldots, T_n)$ with, for $l \leq i \leq n$, $T_i = (\kappa_i, F_i)$ and $\kappa_i = (A, \Sigma, \emptyset, P_i)$. Enumerate $\Sigma = \{\sigma_1, \ldots, \sigma_m\}$. We will use $nm$ by $nm$ diagonal block matrices to represent the current contents of the tubes in the system. If $\boldsymbol{L} = (L_1, L_2, \ldots, L_n)$ is an $n$-tuple of end-marked languages, all subsets of $\Sigma A^* \Sigma$, then the matrix representation of $\boldsymbol{L}$ is

$$M_{\boldsymbol{L}} = \begin{bmatrix} B_1 & 0 & \ldots & 0 \\ 0 & B_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & B_n \end{bmatrix}$$

where each block $B_i$ is an $m$ by $m$ block ($m = |\Sigma|$) which has entries identical to the matrix form of the end-marked language $L_i$, i.e. the $(j,k)$-th position in the $i$-th diagonal block contains the $(\sigma_j, \sigma_k)$ component of $L_i$, and 0 is an $m$ by $m$ block which has entries all $\emptyset$, the empty set.

Similarly, we have a block diagonal *pasting matrix* $P$ the same size whose $i$-th diagonal block corresponds to the end-marked language $P_i$ of pasting rules for the $i$-th test-tube.

Finally for each pair $(\sigma_j, \sigma_k) \in F_i$ we define a pair of block diagonal *moving matrices*

$$U_{i,j,k} = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ I_{j,j} & I_{j,j} & \ldots & I_{j,j} & I_{j,j} \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } V_{i,j,k} = \begin{bmatrix} 0 & \ldots & I_{k,k} & \ldots & 0 \\ 0 & \ldots & I_{k,k} & \ldots & 0 \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ 0 & \ldots & I_{k,k} & \ldots & 0 \\ 0 & \ldots & I_{k,k} & \ldots & 0 \end{bmatrix}$$

where the $I_{j,j}$'s are in the $i$-th row of blocks in $U_{i,j,k}$ and the $I_{k,k}$'s are in the $i$-th column of blocks in $V_{i,j,k}$ and $I_{x,x}$ is a $m$ by $m$ block which has entries all 0 except for the $(x,x)$ entry which is 1. Then for any block diagonal matrix $X$, the block diagonal matrix $U_{i,j,k} X V_{i,j,k}$ has all entries 0 except for the $(j,k)$-th position in the $i$-th diagonal block. This position contains the union of all the $(j,k)$-th positions in all the diagonal blocks of $X$. These moving matrices will be used to implement filtering between test-tubes.

Our proof finishes in a series of three lemmas.

**Lemma 5.2.** *Given $\tau$, a PFH-system as above and matrices $P$, $U_{i,j,k}$ and $V_{i,j,k}$ defined as above from $\tau$, let*

$$\Theta(X) = X + XPX + \sum_{\substack{(\sigma_j, \sigma_k) \in F_i \\ 1 \leq i \leq n}} U_{i,j,k} X V_{i,j,k}$$

*for $X$ a diagonal block matrix. Then the action of $\Theta$ corresponds to the action of $\tau$, i.e. for any tuple of end-marked languages $\boldsymbol{L} \in (\Sigma A^* \Sigma)^n$ we have $\Theta(M_{\boldsymbol{L}}) = M_{\tau(\boldsymbol{L})}$.*

*Proof.* This is just a matter of translating to and from the matrix form for end-marked languages. ☐

Thus to study $\tau^*(\boldsymbol{M})$ we are interested in solving the matrix equation $X = \Theta(X)$ subject to the condition that $M_{\boldsymbol{M}} \subset X$ and the minimal solution to this equation is $\Theta^*(\boldsymbol{M})$.

**Lemma 5.3.** *Let $M$, $P$, $U_l$ and $V_l$ $(1 \leq l \leq t)$ be square matrices of languages. Define $\phi(X) = X + XPX + \sum_{l=1}^{t} U_l X V_l$, define $G$ to be the context free grammar which has a single nonterminal $S$, terminals $\{\mathcal{M}, \mathcal{P}\} \cup \{\mathcal{U}_l, \mathcal{V}_l\}_{l=1}^{t}$ and productions $S \to \mathcal{M}$, $S \to S\mathcal{P}S$ and $S \to \mathcal{U}_l S \mathcal{V}_l$ (for $1 \leq l \leq t$) and define $\psi$ by extending $\psi(\mathcal{M}) = M$, $\psi(\mathcal{P}) = P$, $\psi(\mathcal{U}_l) = U_l$, $\psi(\mathcal{V}_l) = V_l$ for $1 \leq l \leq t$. Then,*

$$\phi^*(M) = \bigcup_{w \in L(G)} \psi(w)$$

*Proof.* Inductively, iterating $\phi$ $k$ times creates a sum whose terms are all the products corresponding to those words which can be derived from $S$ in $k$ or fewer steps. ☐

Finally to show that $\bigcup_{w \in L(G)} \psi(w)$ is context-free we have:

**Lemma 5.4.** *Let $A$ and $B$ be disjoint alphabets, $L \in A^*$ context-free and let $\gamma$ associate with each $a \in A$ an $l$ by $l$ matrix $\gamma(a)$ whose components are all regular subsets of $B^*$. Then $\sum_{w \in L} \gamma(w)$ is context free (where for $w = a_1 a_2 \ldots a_n$ we let $\gamma(w) = \gamma(a_1)\gamma(a_2)\ldots\gamma(a_n)$).*

*Proof.* We will mimic the standard proof that the family of CF languages (or any full trio) is closed under substitution of regular languages [6]. Let $h_A : (A \cup B)^* \to A^*$ be the extension of the identity on $A$ and the trivial map on $B$. Similarly, let $h_B : (A \cup B)^* \to B^*$ be the extension of the identity on $B$ and the trivial map on $A$. For each $a \in A$, let $D_a$ be the $l$ by $l$ matrix with $a$'s on the main diagonal and 0's off the diagonal. Let $D = \bigcup_{a \in A} D_a \gamma(a)$ (so that $D_{i,j} = \{aw \mid w \in \gamma(a)_{i,j}\}$). Then $D$ is a finite union of regular matrices and therefore regular. Let $U$ be an $l$ by $l$ matrix all of whose entries are $h_A^{-1}(L)$. We will show $\bigcup_{w \in L} \gamma(w) = h_B(U \cap D^*)$ and will thus be CF since it is the homomorphic image of the intersection of a CF matrix and a regular matrix.

First we show $\bigcup_{w \in L} \gamma(w) \subset h_B(U \cap D^*)$. Let $b_1 b_2 \ldots b_n \in \gamma(w)_{i,j}$ for some $w \in L$. Then write $w = a_1 a_2 \ldots a_k$ and we have $b_1 b_2 \ldots b_n \in [\gamma(a_1)\gamma(a_2)\ldots\gamma(a_k)]_{i,j}$ so that $b_1 b_2 \ldots b_n = w_1 w_2 \ldots w_k$ where $w_1 \in \gamma(a_1)_{i,l_1}$, $w_m \in \gamma(a_m)_{l_{m-1},l_m}$ for $1 < m < k$ and $w_k \in \gamma(a_k)_{l_{k-1},j}$. Then we have

$$a_1 w_1 a_2 w_2 \ldots a_k w_k \in [a_1 \gamma(a_1) \ldots a_k \gamma(a_k)]_{i,j} \subset D^*_{i,j}.$$

Furthermore, we have $h_A(a_1 w_1 a_2 w_2 \ldots a_k w_k) = a_1 a_2 \ldots a_k \in L$ so that we have

$$a_1 w_1 a_2 w_2 \ldots a_k w_k \in (U \cap D^*)_{i,j}$$

and

$$b_1 b_2 \ldots b_n = w_1 w_2 \ldots w_k = h_B(a_1 w_1 a_2 w_2 \ldots a_k w_k) \in h_B(U \cap D^*)_{i,j}$$

Next we show $h_B(U \cap D^*) \subset \bigcup_{w \in L} \gamma(w)$. Let $b_1 b_2 \ldots b_n \in h_B(U \cap D^*)_{i,j}$ then $b_1 b_2 \ldots b_n = h_B(u)$ for some $u \in (U \cap D^*)_{i,j}$. Then $u \in U_{i,j} = h_A^{-1}(L)$ so that $h_A(u) = a_1 a_2 \ldots a_k \in L$ and

$u \in D^*_{i,j}$, and we must have $u = a_1u_1a_2u_2 \dots a_ku_k$ where $u_1 \in \gamma(a_1)_{i,l_1}$, $u_m \in \gamma(a_m)_{l_{m-1},l_m}$ for $1 < m < k$ and $u_k \in \gamma(a_k)_{l_{k-1},j}$ so that

$$b_1b_2 \dots b_n = h_B(u) = u_1u_2 \dots u_k \in [\gamma(a_1)\gamma(a_2) \dots \gamma(a_k)]_{i,j}$$

and since $a_1a_2 \dots a_k \in L$ we have $b_1b_2 \dots b_n \in \gamma(a_1a_2 \dots a_k) \in [\bigcup_{w \in L} \gamma(w)]_{i,j}$. $\square$

These three lemmas finish the proof of Proposition 3.1.

## 6. Equivalence of Various Multi-Test-Tube Approaches

The traditional formal model of DNA computing as laid out by Head [5] used a splicing operation instead of the cutting and pasting we have been doing. Here is one standard formulation of splicing. Let $A$ be an alphabet. Then a *splicing rule* is a tuple $(u, v; u', v')$ where $u$, $v$, $u'$ and $v'$ are strings in $A^*$. Given a splicing rule $r = (u, v; u', v')$ and strings $w = xuu'y'$ and $w' = x'v'vy$ we say *the result of splicing* $w$ *and* $w'$ *using the splicing rule* $r$ *is* $xuvy$. A *splicing scheme* or *H-scheme* is a pair $\sigma = (A, R)$ where $A$ is an alphabet and $R$ is a set of splicing rules. Given a language $L \subset A^*$ and an H-scheme $\sigma = (A, R)$, we define $\sigma(L)$ to be the union of $L$ and the set of all strings that can be obtained by splicing two strings of of $L$ using a rule in $R$. We write $\sigma^0(L) = L$. For each integer $i > 0$ we define $\sigma^i(L) = \sigma(\sigma^{i-1}(L))$ and we let $\sigma^*(L) = \bigcup_{i=0}^{i=\infty} \sigma^i(L)$. An *H-system* is a pair $(M, \sigma)$ where $\sigma$ is an H-scheme and $M$ is a set of strings called the *initial molecules* of the H-system. Pixton and others have shown that, in a single test-tube, splicing, cutting and pasting and cutting and recombination are essentially equivalent[9, 4]. We wish to compare the various types of filtering mechanisms that have been described for test-tube systems.

Păun and others [2, 7] have described test-tube systems based on splicing in which filtering between tubes is by intersection with $F^*$ where $F$ is a subset of the alphabet $A$. Formally: an *n test-tube* $F^*$*- filtered H-scheme (nTTH or simply TTH-scheme)* is a tuple $\tau = (A, T_1, \dots, T_n)$ where each $T_i$ is a called a *test-tube*, and each $T_i = (\sigma_i, F_i)$ where $\sigma_i = (A, R_i)$ is an H-scheme and $F_i \subset A$. We call $F_i$ the *filter* for tube $T_i$. Given an $n$-tuple of end-marked languages $\boldsymbol{L} = (L_1, L_2, \dots, L_n)$ we define the action of $\tau$ on $\boldsymbol{L}$ by component as

$$(\tau(\boldsymbol{L}))_i = (\kappa^*_i(L_i) - \bigcup_{j=1}^{n}(\kappa^*_i(L_i) \cap F^*_j)) \cup \bigcup_{j=1}^{n}(\kappa^*_j(L_j) \cap F^*_i).$$

Define $\tau^0(\boldsymbol{L}) = \boldsymbol{L}$ and $\tau^{j+1}(\boldsymbol{L}) = \tau(\tau^j(\boldsymbol{L}))$. Thus, $\tau^j(\boldsymbol{L})$ represents the contents of the tubes after $j$ computation and filtering cycles. Note that, with the filtering set up in this fashion, it is not always the case that $L_i$ will be a subset of $(\tau(\boldsymbol{L}))_i$. Păun and others define the *result* of the system is to be $\cup_{j=0}^{\infty}(\tau^j(\boldsymbol{L}))_1$. Below we will discuss the differences between this multi-test-tube computing system and ours.

First, we address some technical issues concerning the sequence in which filtering and other test-tube actions occur. We will refer to these other actions (cutting and pasting or splicing) as computations. For this discussion we will focus on the various orders in which filtering and computation occur. To do so we will modify our notation. We will work within the end-marker filtered cutting and pasting context. In our definition of a CPFH-scheme, given a tuple $\tau$ we defined the action of $\tau$ on a tuple of end-marked languages $\boldsymbol{L}$. We now want to consider different ways to define the action of such a tuple $\tau$ on a tuple of end-marked languages. Thus, we will now denote our action as $G(\tau, \boldsymbol{L})$ so that:

$$(G(\tau, \boldsymbol{L}))_i = \kappa_i(L_i) \cup \bigcup_{j=1}^{j=n} (\kappa_j(L_j) \cap L(F_i))$$

The action described by Păun and others [10, 1, 2] translates in our (end-marker filtered, cut and paste) context to something we will denote by $\hat{G}(\tau, \boldsymbol{L})$ and it is defined as follows:

$$(\hat{G}(\tau, \boldsymbol{L}))_i = (\kappa_i^*(L_i) - \bigcup_{j=1}^{n} (\kappa_i^*(L_i) \cap L(F_j))) \cup \bigcup_{j=1}^{n} (\kappa_j^*(L_j) \cap L(F_i)).$$

There are two differences between these two formulations of multi-test-tube filtering. In $G$ there is only one iteration of the $\kappa_i$'s between rounds of filtering. In $\hat{G}$ there is an unlimited number of iterations; $\kappa_i$ has been replaced by $\kappa_i^*$. (In any concrete implementation, of course, the reality would probably be something in between.) Also, in $G$ copies of all strings produced in test-tube $i$ remain in test-tube $i$ as well as being filtered into other tubes. In $\hat{G}$ strings which are produced in test-tube $i$ and pass a filter for other tubes are removed from test-tube $i$ unless they pass the filter for test-tube $i$ as well. Note that in both $G$ and $\hat{G}$, a string may pass the filter for more than one test-tube, in which case copies of that string are placed in all of the test-tubes for which it passes the filter.

Let $G^0(\tau, \boldsymbol{L}) = \boldsymbol{L}$, $G^{j+1}(\tau, \boldsymbol{L}) = G(\tau, G^j(\tau, \boldsymbol{L}))$, and $G^*(\tau, \boldsymbol{L}) = \cup_{j=0}^{\infty} G^j(\tau, \boldsymbol{L})$. We then define the *full result* of the $G$ system to be $\gamma(\tau, \boldsymbol{L}) = \text{Strip}((G^*(\tau, L))_1)$ and the $(\alpha, \beta)$*-result* of the $G$ system to be $\text{Strip}((G^*(\tau, L))_1 \cap \alpha A^* \beta)$. We wish to similarly define the result of $\hat{G}$. We define $\hat{G}^0(\tau, \boldsymbol{L}) = \boldsymbol{L}$ and $\hat{G}^{j+1}(\tau, \boldsymbol{L}) = \hat{G}(\tau, \hat{G}^j(\tau, \boldsymbol{L}))$. We then define the *full result* of the $\hat{G}$ system to be $\hat{\gamma}(\tau, \boldsymbol{L}) = \text{Strip}(\cup_{j=0}^{\infty} (\hat{G}^j(\tau, L))_1)$. Further, since we are using end-markers, we can define the $(\alpha, \beta)$*-result* of the $\hat{G}$ system to be $\text{Strip}(\cup_{j=0}^{\infty} (\hat{G}^j(\tau, L))_1 \cap \alpha A^* \beta)$.

We will argue that these two formulations have the same generative capacity. The creation of any particular string in the result of either type of system involves only a finite number of cuts, pastes and filtering steps. If we start both systems with the same initial contents and the same cutting, pasting and filtering rules, then, provided all the needed ingredient strings are present, the difference between a derivation of a result string using a single iteration of the $\kappa$'s between filtering steps and the same derivation using $\kappa^*$'s between filtering steps is a matter of timing. However we need to be careful to make sure that needed ingredient strings become available and/or remain available. Differences can occur since, in the $\hat{G}$-systems, test-tubes do not always retain copies of all of their contents after filtering. The following two lemmas show how to handle these issues.

**Lemma 6.1.** *Let* $\tau = (A, \Sigma, T_1, \ldots, T_n)$ *be a tuple as in our definition of CPFH-system and let* $\boldsymbol{L} \subset (\Sigma A^* \Sigma)^n$ *be a tuple of end-marked languages. Then there exists a tuple* $\hat{\tau} = (A, \hat{\Sigma}, \hat{T}_1, \ldots, \hat{T}_n)$ *and a tuple of end-marked languages* $\hat{\boldsymbol{L}} \subset (\hat{\Sigma} A^* \hat{\Sigma})^n$ *such that* $\gamma(\tau, \boldsymbol{L}) = \hat{\gamma}(\hat{\tau}, \hat{\boldsymbol{L}})$*. Furthermore, each set associated with* $\tau$ *and* $\boldsymbol{L}$ *(i.e. each of* $\Sigma$ *and the* $C_i, P_i, F_i, L_i$*) differs from the corresponding set associated with* $\hat{\tau}$ *and* $\hat{\boldsymbol{L}}$ *by a finite set. Finally, if there is no cutting in* $\tau$ *there will be no cutting in* $\hat{\tau}$*.*

*Proof.* Define $\hat{\tau}$ and $\hat{\boldsymbol{L}}$ as follows. For each $\alpha \in \Sigma$ and for each $1 \leq i \leq n$ find two new symbols (not in $A \cup \Sigma$) denoted $\alpha_i$ and $\bar{\alpha}_i$. Then define

$$\hat{\Sigma} = \Sigma \cup \{\alpha_i, \bar{\alpha}_i \mid \alpha \in \Sigma, 1 \leq i \leq n\}.$$

Further, for each $1 \leq i \leq n$ define

$$\hat{L}_i = L_i \cup \{\bar{\alpha}_i\alpha_i, \alpha\alpha_i, \alpha_i\bar{\alpha}_i, \alpha_i\alpha \mid \alpha \in \Sigma\}, \hat{F}_i = F_i \cup \{(\bar{\alpha}_i, \bar{\beta}_i) \mid \alpha, \beta \in \Sigma\},$$

$$\hat{C}_i = C_i, \text{ and } \hat{P}_i = P_i \cup \{\alpha_i\alpha, \alpha_i\bar{\alpha}_i, \alpha\alpha_i, \bar{\alpha}_i\alpha_i \mid \alpha \in \Sigma\}.$$

The new end-markers, the additions to $\boldsymbol{L}$ and the new pasting rules together create extra copies of each string created in each tube. If $\alpha w\beta$ exists in test-tube $\hat{T}_i$ at some point then $\bar{\alpha}_i w\beta$, $\alpha w\bar{\beta}_i$, and $\bar{\alpha}_i w\bar{\beta}_i$ exist in that tube as well. $\bar{\alpha}_i w\bar{\beta}_i$ passes the filter $\hat{F}_i$ so it will remain in tube $\hat{T}_i$ after a filtering step. The pasting rules will then recreate $\alpha w\beta$ so that it will be available as needed to duplicate the action of $\tau$. In the end, while $(G^*(\tau, \boldsymbol{L}))_i \neq \cup_{j=0}^{j=\infty}(\hat{G}^j(\hat{\tau}, \hat{\boldsymbol{L}}))_i$, the difference is just the extra copies of strings with the new barred end-markers. The difference disappears when the end-markers are stripped. In fact, for $\alpha$ and $\beta$ in the original $\Sigma$, the $(\alpha, \beta)$-results are the same, as is the full result. ☐

**Lemma 6.2.** *Let* $\hat{\tau} = (A, \hat{\Sigma}, \hat{T}_1, \ldots, \hat{T}_{\hat{n}})$ *be a tuple as in our definition of CPFH-system and let* $\hat{\boldsymbol{L}} \subset (\hat{\Sigma}A^*\hat{\Sigma})^n$ *be a tuple of end-marked languages. Then there exists a tuple* $\tau = (A, \Sigma, T_1, \ldots, T_n)$ *and a tuple of end-marked languages* $\boldsymbol{L} \subset (\Sigma A^*\Sigma)^n$ *such that* $\hat{\gamma}(\hat{\tau}, \hat{\boldsymbol{L}}) = \gamma(\tau, \boldsymbol{L})$*. Furthermore,* $n > \hat{n}$ *and as before, each set associated with* $\tau$ *differs from the corresponding set associated with* $\hat{\tau}$ *by a finite set. Finally, if there is no cutting in* $\hat{\tau}$ *there will be no cutting in* $\tau$*.*

*Proof.* In $\hat{G}$, after each round of filtering the contents of a tube consist of strings which either pass the filter for that tube or pass the filter for no tube. However, anything found in a tube after a round of filtering but before more computation begins will remain in the tube through all subsequent rounds of computation and filtering. Thus, after the first round of filtering, anything created in a tube during a round of computation will be recreated in all subsequent rounds of computation. Thus, our only concern is with those elements of the initial contents $(\hat{\boldsymbol{L}})_i$ that pass the filter for some other test-tube $\hat{T}_j$ but do not pass the filter $\hat{F}_i$. Such strings are available in test-tube $\hat{T}_i$ only before the first round of filtering. For simplicity of definition, we define $\tau$ with $n = 2\hat{n} + 1$ tubes (two tubes is $\tau$ for each tube of $\hat{\tau}$ and a final collection tube) but the duplicate tubes are really needed only for those $i$ for which some elements of $(\hat{\boldsymbol{L}})_i$ will be removed with the first round of filtering. Define $\tau$ and $\boldsymbol{L}$ as follows. For each $\alpha \in \hat{\Sigma}$ and $1 \leq i \leq \hat{n}$ find two new symbols (not in $A \cup \hat{\Sigma}$) denoted $\alpha_i$ and $\bar{\alpha}_i$. Then define

$$\Sigma = \hat{\Sigma} \cup \{\alpha_i, \bar{\alpha}_i \mid \alpha \in \hat{\Sigma}, 1 \leq i \leq \hat{n}\}.$$

Further, for each $1 \leq i \leq \hat{n}$ let

$$\begin{aligned}
L_i &= (\hat{L}_i - \cup_{j=1}^{j=n} L(F_j)) \cup (\hat{L}_i \cap L(F_i)) \cup \{\alpha\alpha_i, \alpha_i\alpha \mid \alpha \in \hat{\Sigma}\}, \\
L_{\hat{n}+i} &= \hat{L}_i \cup \{\bar{\alpha}_i\alpha_i, \alpha_i\bar{\alpha}_i \mid \alpha \in \hat{\Sigma}\}, \\
F_i &= \hat{F}_i \cup \{(\bar{\alpha}_i, \bar{\beta}_i) \mid (\alpha, \beta) \notin \cup_{j=1}^{j=n} F_j\}, \\
F_{\hat{n}+i} &= \emptyset, \\
C_i &= C_{\hat{n}+i} = \hat{C}_i \\
P_i &= \hat{P}_i \cup \{\alpha_i\bar{\alpha}_i, \bar{\alpha}_i\alpha_i \mid \alpha \in \hat{\Sigma}\} \text{ and} \\
P_{\hat{n}+i} &= \hat{P}_i \cup \{\alpha_i\alpha, \alpha\alpha_i \mid \alpha \in \hat{\Sigma}\}.
\end{aligned}$$

Test-tube $\hat{n} + i$ produces eventually all strings that are produced before the first round of filtering in $(\hat{G}(\hat{\tau}, \hat{\boldsymbol{L}}))_i$. The strings are produced in the form $\alpha w \beta$ and then, as in previous proofs, the end-markers are changed to produce the form $\bar{\alpha}_i w \bar{\beta}_i$. Those $\bar{\alpha}_i w \bar{\beta}_i$ strings which would have remained in $\hat{T}_i$ after the first round of filtering are then filtered into test-tube $T_i$ where the end-markers are changed back and the $\alpha w \beta$ forms will be available for use. Thus, for $1 \leq i \leq \hat{n}$, we have $\text{Strip}((G^*(\tau, \boldsymbol{l}))_i)$ almost equal to $\text{Strip}(\cup_{j=0}^{j=\infty}(\hat{G}^j(\hat{\tau}, \hat{\boldsymbol{L}}))_i)$: missing are those strings in $\hat{L}_i$ which are removed by the first round of filtering. To exactly duplicate the results of of the $\hat{G}$ system, we need one last tube $T_0$ which will contain the entire initial contents of the first tube, $\hat{L}_1$, and collect everything occurring in tube $T_1$, but with no computation occurring in tube $T_0$. To do this, we must create special copies of the strings in tube 1 to filter into $T_0$. We will need a few more end-markers, so add to $\Sigma$ two new symbols (not in $A \cup \Sigma$ before) $\Delta$ and $\xi$. Add to $L_1$ the strings $\xi\Delta$ and $\Delta\xi$. Add to $P_1$ all of the strings $\xi\alpha$ and $\alpha\xi$ for $\alpha \in \Sigma - \{\Delta, \xi\}$. This will create in $T_1$ a string of the form $\Delta w \Delta$ for each string $\alpha w \beta$ already present in $T_1$. Let $L_0 = \hat{L}_1$, $F_0 = \{(\Delta, \Delta)\}$. Then we will have

$$\text{Strip}((G^*(\tau, \boldsymbol{L}))_0) = \text{Strip}((\cup_{j=0}^{j=\infty} \hat{G}^j(\hat{\tau}, \boldsymbol{L}))_1),$$

i.e. the stripped result of the new system (in tube 0) will be the same as the result of the original. □

Now, we illustrate two techniques for using end-marker filtering to model filtering with filters of the form $F^*$. The first is the technique used in section 4 and the second is one that does not involve cutting. For the first technique we illustrate only filtering between two tubes, but the technique could be used repeatedly in a more complex system to replace *-filtering by end-marker filtering.

**Lemma 6.3.** *Let $A$ be a finite set, $L \subset A^*$ and $F \subset A$. Then there is a tuple $\tau = (A', \Sigma, T_1, T_2, T_3)$ and end-markers $\epsilon$, $\delta$ and $\Delta$ not in $A$, so that $\gamma(\tau, \boldsymbol{L}) = \hat{\gamma}(\tau, \boldsymbol{L}) = L \cap F^*$ where $\boldsymbol{L} = (\emptyset, \{\epsilon\delta, \delta\epsilon\}, \Delta L \Delta \cup \{\epsilon\delta, \delta\epsilon\})$.*

*Proof.* In addition to $\delta$, $\epsilon$ and $\Delta$, find symbols $l$, $B$, and $r$ and end-markers $\nabla$ and (for each $a \in F$) $\gamma_a$ and $\sigma_a$ not in $A$. Then let $A' = A \cup \{l, B, r\}$ and $\Sigma = \{\Delta, \nabla, \delta, \epsilon\} \cup \{\gamma_a, \sigma_a \mid a \in F\}$.

Define the rest of $\tau$ as follows:

$$\begin{aligned}&F_1 = \{(\nabla,\nabla)\}, C_1 = P_1 = \emptyset\\ &F_2 = \{(\gamma_a,\sigma_a) \mid a \in F\}, C_2 = \emptyset, P_2 = \{\epsilon l a\gamma_a, \sigma_a r\epsilon \mid a \in F\}\\ &F_3 = \{(\Delta,\Delta),(\epsilon,\epsilon)\}, C_3 = \{\epsilon l\nabla, \nabla Br\epsilon\} \cup \{\epsilon l\gamma_a, \sigma_a ar\epsilon \mid a \in F\}\\ &\text{and } P_3 = \{\epsilon lB\Delta, \Delta r\epsilon\}.\end{aligned}$$

For each $w \in L$, $\Delta w\Delta$ exists in the third test-tube and pasting there creates $\epsilon lBwr\epsilon$. Tubes 2 and 3 work together to rotate letters in $F$. Only if $w$ is in $F^*$ will it be completely rotated to form $\epsilon lwBr\epsilon$. Only those words are cut to form $\nabla w\nabla$ which is filtered into the first tube. As before, we could obtain $\nabla L\nabla$ as a component-result using only two tubes. ☐

Note that in the above construction, the cutting rules all use the new symbols $l$, $r$, and $B$ and the pasting rules all involve only the new end-markers so that this construction can be added to an existing test-tube system without interference between the new cutting and pasting used for filtering and the existing cutting and pasting.

If we are interested in restating our context free results in the setting of traditional multi-test-tube splicing, we need a method of modeling *-filtering in PF-systems, i.e. without cutting. The next lemma illustrates such a method, showing how to filter the contents of one pasting-only tube into other tubes. Again to implement more complex test-tube systems this technique would need to be used over and over.

**Lemma 6.4.** *Let $A$ and $\Sigma$ be finite sets and let $F_i \subset A$ for each $1 \leq i \leq n$. Further, let $(M,\kappa)$ be a CPH-system with no cutting (where $\kappa = (A,\Sigma,\emptyset,P)$). Then there is a tuple of test tubes $\tau = (A,\Sigma',T_1,T_2,\dots,T_{n+1})$ with no cutting in any tube and pasting only in $T_{n+1}$, a tuple of initial contents $\boldsymbol{M} = (\emptyset,\dots,\emptyset,M')$ and a map $\phi : A\cup\Sigma' \to A\cup\Sigma$ which is the identity on $A$ (extending to $(A\cup\Sigma')^*$) so that $\phi((G(\tau,\boldsymbol{M}))_i) = \phi((\hat{G}(\tau,\boldsymbol{M}))_i) = \kappa^*(M)\cap\Sigma F_i^*\Sigma$ where $\Sigma'$ is finite, $M'$ is finite if $M$ is finite and $P_{n+1}$ is finite if $P$ is finite.*

*Proof.* We will use $n$ copies of the end-markers in $\Sigma$. For each $\sigma \in \Sigma$ and for each $1 \leq i \leq n$ find a new end-marker $\sigma_i$. Let $\Sigma_i = \{\sigma_i, \mid \sigma \in \Sigma\}$ and let $\Sigma' = \cup_{i=1}^{i=n}\Sigma_i$. For each $1 \leq i \leq n$ let $T_i = (\kappa_i, \Sigma_i\times\Sigma_i)$ where $\kappa_i = (A,\Sigma_i,\emptyset,\emptyset)$. Let $T_{n+1} = (\kappa_{n+1},\Sigma'\times\Sigma')$ where $\kappa_{n+1} = (A,\Sigma',\emptyset,P_{n+1})$ and $P_{n+1} = \cup_{i=1}^{i=n}\{\alpha_i w\beta_i \mid \alpha w\beta \in P\cap\Sigma F_i^*\Sigma\}$. Finally let $M' = \cup_{i=1}^{i=n}\{\alpha_i w\beta_i \mid \alpha w\beta \in M\cap\Sigma F_i^*\Sigma\}$ and define $\phi(\sigma_i) = \sigma$ for each $\sigma_i \in \Sigma'$. Since there is no cutting in $\kappa$, any element $w \in \mathrm{Strip}(\kappa^*(M))\cap F_i^*$ must be assembled by $\kappa$ from elements of $M\cap\Sigma F_i^*\Sigma$ pasted together using elements of $P\cap\Sigma F_i^*\Sigma$. The same pastings will occur in $T_{n+1}$ but with end-markers in $\Sigma_i$ instead of $\Sigma$ so the result will be filtered into $T_i$. Furthermore, only such pastings will produce results that will be filtered into $T_i$. ☐

Now we look at a way to model end-marker filtering using *-filtering, again in the context of pasting alone.

**Lemma 6.5.** *Given a CPH-scheme $\kappa = (A,\Sigma,\emptyset,P)$ with no cutting, a set of initial test-tube contents $L \subset \Sigma A^*\Sigma$ and $F \subset \Sigma\times\Sigma$, there exists another CPH-scheme $\overline{\kappa} = (A,\overline{\Sigma},\emptyset,\overline{P})$, another set of initial tube contents $\overline{L}$, and, for each $(\alpha,\beta)\in F$, a filter set $\overline{F}_{(\alpha,\beta)} \subset A\cup\overline{\Sigma}$ and a map $\phi : A\cup\overline{\Sigma} \to A\cup\Sigma$ which is the identity on $A$ (extending to $(A\cup\overline{\Sigma})^*$) so that $\kappa^*(L)\cap L(F) = \phi(\bigcup_{(\alpha,\beta)\in F}(\overline{\kappa}^*(\overline{L})\cap\overline{F}^*_{(\alpha,\beta)}))$. Further, $\overline{L}$ is finite if $L$ is finite and $\overline{P}$ is finite if $P$ is finite.*

*Proof.* To mimic the effect of the ordered pairs in $F$ we need two copies of the end-markers in $\Sigma$. Thus, we let $\overline{\Sigma} = \{\sigma_L, \sigma_R \mid \sigma \in \Sigma\}$. Let $\overline{L} = \{\alpha_L w \beta_R \mid \alpha w \beta \in L\}$, $\overline{P} = \{\alpha_R w \beta_L \mid \alpha w \beta \in P\}$, and $\overline{F}_{(\alpha,\beta)} = A \cup \{\alpha_L, \beta_R\}$. Let $\phi(\sigma_L) = \phi(\sigma_R) = \sigma$ for each $\sigma \in \Sigma$. Let $X = \{\sigma_L \mid \sigma \in \Sigma\}A^*\{\sigma_R \mid \sigma \in \Sigma\}$. Then $\overline{L} \subset X$ and $\overline{\kappa}(X) \subset X$ so that $\overline{\kappa}^*(\overline{L}) \subset X$. Thus, for each $(\alpha, \beta) \in F$ we have the $(\alpha, \beta)$ component of $\kappa^*(L)$ equal to $\phi(\overline{\kappa}^*(\overline{L}) \cap \overline{F}_{(\alpha,\beta)})$. □

This gives us what we need to mimic end-marker filtering with *-filtering. In the context of the lemma above, to mimic filtering the results of $\kappa(L)$ with the end-marker filter $F$ into a new tube, we proceed as follows. Set up $|F|$ test-tubes denoted $T_{(\alpha,\beta)}$ for $(\alpha, \beta) \in F$ and a final collection tube $T$. Filter into $T_{(\alpha,\beta)}$ using *-filtering on the set $\overline{F}_{(\alpha,\beta)}$ and then filter into the collection tube from these tubes using *-filtering on the set $A \cup \overline{\Sigma}$. The final result will be to mimic end-marker filtering from the original tube to $T$ using the filter $F$ with a system of *-filtering.

The last piece needed to restate our context free results in the setting of traditional multi-test-tube splicing is a way to translate from splicing to cutting and pasting and back. Pixton [9] has investigated these conversions in great detail and we summarize his work here for our special case of pasting but not cutting in two lemmas.

**Lemma 6.6.** *Let* $(M, \kappa)$ *be a CPH-system without cutting (so,* $\kappa = (A, \Sigma, \emptyset, P)$*). Then there is another CPH-system* $(\overline{M}, \overline{\kappa})$ *with* $\overline{\kappa} = (A, \overline{\Sigma}, \emptyset, \overline{P})$ *so that* $\overline{P} \subset \overline{\Sigma}\overline{\Sigma}$ *(i.e. there is only empty pasting) and* $\kappa^*(M) = \Sigma A^* \Sigma \cap \overline{\kappa}^*(\overline{M})$*. Further* $\overline{M}$ *is finite if* $M$ *and* $P$ *are finite.*

*Proof.* We will add our pasting strings to the initial tube contents (with modified end-markers identifying them) as follows. We need modified copies of the end-markers, so we let $\overline{\Sigma} = \Sigma \cup \{\overline{\alpha} \mid \alpha \in \Sigma\}$. Let $\overline{M} = M \cup \{\overline{\alpha} w \overline{\beta} \mid \alpha w \beta \in P\}$ and let $\overline{P} = \{\alpha\overline{\alpha}, \overline{\beta}\beta \mid \alpha w \beta \in P\}$. The pasting of $\epsilon w_1 \alpha$ and $\beta w_2 \delta$ using pasting string $\alpha w \beta$ in $(M, \kappa)$ to create $\epsilon w_1 w w_2 \delta$ is replaced in $(\overline{M}, \overline{\kappa})$ by pasting $\epsilon w_1 \alpha$, $\overline{\alpha} w \overline{\beta}$ and $\beta w_2 \delta$ using the two pasting strings $\alpha\overline{\alpha}$ and $\overline{\beta}\beta$. Conversely, the form of the pasting rules ensures that any string in $\overline{\kappa}^*(\overline{M})$ was created by pasting together an alternating list of strings from the old $M$ and $\{\overline{\alpha} w \overline{\beta} \mid \alpha w \beta \in P\}$ so that any string in $\Sigma A^* \Sigma \cap \overline{\kappa}^*(\overline{M})$ is also created in $\kappa^*(M)$ by pasting those strings in the list which are from $M$ to each other using pasting rules corresponding to those strings in the list from $\{\overline{\alpha} w \overline{\beta} \mid \alpha w \beta \in P\}$. □

**Lemma 6.7.** *Let* $(M, \kappa)$ *be a CPH-system where* $\kappa = (A, \Sigma, \emptyset, P)$ *and* $P \subset \Sigma\Sigma$*. If we let* $\sigma$ *be the H-scheme* $(A \cup \Sigma, \{(1, \alpha; \beta, 1) \mid \alpha\beta \in P\})$ *then* $\kappa^*(M) = \sigma^*(M)$*.*

*Proof.* The pasting of $\epsilon w_1 \alpha$ and $\beta w_2 \delta$ using pasting string $\alpha\beta$ in $(M, \kappa)$ to create $\epsilon w_1 w_2 \delta$ is equivalent to the splicing of $\epsilon w_1 \alpha$ and $\beta w_2 \delta$ using splicing rule $(1, \alpha; \beta, 1)$ in $(M, \sigma)$ to create $\epsilon w_1 w_2 \delta$. □

Others have shown that TTH-systems can generate any RE language [1, 10], analogous to Proposition 3.1. Here we finally have a TTH result analogous to Proposition 3.2.

**Proposition 6.1.** *Let* $\tau = (A \cup \Sigma, T_1, \ldots, T_n)$ *(*$T_i = (\sigma_i, F_i)$ *and* $\sigma_i = (A \cup \Sigma, R_i)$*) be a finite TTH-scheme where* $A \cap \Sigma = \emptyset$ *and each splicing rule in each* $R_i$ *is of the form* $(1, \alpha; \beta, 1)$ *for some* $\alpha, \beta \in \Sigma$*. Further, let* $\boldsymbol{M}$ *be an n-tuple of finite initial tube contents (*$\boldsymbol{M} \subset (\Sigma A^* \Sigma)^n$*). Then the result of the TTH-system* $(\tau, \boldsymbol{M})$ *is context free and any context free set can be obtained as the result of a finite TTH-system all of whose splicing rules are of the form* $(1, a; b, 1)$*.*

*Proof.* Given $\tau$ as described in the statement of the lemma, we can create a system of cut and paste test-tubes connected by $F^*$-filters by replacing each H-scheme $\sigma_i$ by a CPH-scheme $\kappa_i = (A, \Sigma, \emptyset, P_i)$ where $P_i = \{\alpha\beta \mid (1, \alpha; \beta, 1) \in R_i\}$. Then as in the proof of Lemma 6.7 we have $\kappa_i^*(L) = \sigma_i^*(L)$ for any $L$ in $\Sigma A^* \Sigma$. Next, using Lemma 6.4 repeatedly, we can replace the $F^*$-filtering by end-marker filtering. Then Lemma 6.2 creates the CPF-system we want: one which has no cutting and has a result the same as the result of $\tau$. Thus Proposition 3.2 tells us the result of $\tau$ must be context free.

Conversely, given a context free set, we can create a PF-system which has that set as its result. Then we replace the end-marker filtering with $F^*$-filtering using repeated applications of the techniques in Lemma 6.5. Finally, we can replace the pasting in each tube with splicing as in Lemma 6.7, creating the TTH-system we want. ☐

Finally, we consider one last variation. When working with $n$-test-tube systems, Freund [3] uses $n^2$ filters, one between each pair of tubes. An end-marker filtering system using these between-tube-filters can be modeled using our tube-entry-filters and extra end-markers, as follows. Consider a pair $(\alpha, \beta)$ in the filter from test-tube $i$ to test-tube $j$. Add pasting rules to test-tube $i$, as we have in the previous proofs, so that each string created in test-tube $i$ which has the form $\alpha w \beta$ creates another string $\alpha_j w \beta_j$ and then add $(\alpha_j, \beta_j)$ to the tube-entry-filter for test-tube-$j$. Furthermore, tube-entry-filters can easily be implemented with between-tube-filters. It is also worth noting that between-tube-filters work very naturally in the matrix formulations of test-tube systems.

## REFERENCES

1. Erzsébet Csuhaj-Varjú, Lila Kari, and Gheorghe Păun, *Test tube distributed systems based on splicing*, Computers and AI **15** (1996), no. 2–3, 211–232.
2. Rudolf Freund, Erzsébet Csuhaj-Varjú, and Franz Wachtler, *Test tube systems with cutting/recombination operations*, Pacific Symposium on Biocomputing, World Scientific Publishing Co., Singapore, 1997.
3. Rudolf Freund and Franzisca Freund, *Test tube systems: when two tubes are enough*, Developments in Language Theory, Aachen, July6–9, 1999, pp. 275–286.
4. Rudolf Freund and F. Wachtler, *Universal systems with operations related to splicing*, Computer and AI **15** (1996), no. 4, 273–293.
5. Thomas Head, *Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors.*, Bulletin of Mathematical Biology **49** (1987), no. 6, 737–759.
6. John E. Hopcroft and Jeffery D. Ullman, *Introduction to automata theory, languages, and computation*, Addison-Wesley, 1979.
7. Gheorghe Păun, *Regular extended H systems are computationally universal*, Journal of Automata, Languages, Combinatorics **1** (1996), no. 1, 27–36.
8. Dennis Pixton, *Regularity of splicing languages*, Discrete Applied Mathematics **69** (1996), no. 1–2, 101–124.
9. ______, *Splicing in abstract families of languages*, Theoretical Computer Science **234** (2000), 135–166.
10. Lutz Priese, Yurii Rogozhin, and Maurice Margenstern, *Finite H-systems with 3 test tubes are not predictable*, Pacific Symposium on Biocomputing (Lawrence Hunter Russ B. Altman, A. Kieth Dunker and Teri Klein, eds.), World Scientific Publishing Co., Singapore, 1998, pp. 547–558.

DEPARTMENT OF MATHEMATICAL SCIENCES, BINGHAMTON UNIVERSITY, BINGHAMTON, NY 13902
*E-mail address*: `peggy@math.binghamton.edu`