Outfix-Guided Insertion

Da-Jung Cho\textsuperscript{1}    Yo-Sub Han\textsuperscript{1}    Timothy Ng\textsuperscript{2}  
Kai Salomaa\textsuperscript{2}

\textsuperscript{1}Department of Computer Science, Yonsei University  
\textsuperscript{2}School of Computing, Queen’s University

DLT 2016, Montréal, QC, Canada
**Input:** A given DNA

**Output:** A desired DNA

**Step 1:** Cut given DNA using primers \(a\) and \(b\)

**Step 2:** Annealing inserted sequence using primers \(c\) and \(d\)

**Step 3:** Ligation PCR with product \(A\), \(B\) and \(C\)
Let $w, x, y, z \in \Sigma^*$. If $w = xyz$, we say $x$ is a prefix of $w$, $z$ is a suffix of $w$, and $(x, z)$ is an outfix of $w$. 
Classical insertion [Haussler 1983]

\[ x \leftarrow y = \{ x_1 y x_2 \mid x = x_1 x_2 \}. \]
Classical insertion [Haussler 1983]

\[ x \leftarrow y = \{ x_1 y x_2 \mid x = x_1 x_2 \}. \]

Contextual insertion [Galiukschov 1981]

\[ x \leftarrow^C y = \{ x_1 uyvx_2 \mid (u, v) \in C, x = x_1 uvx_2 \}. \]
Classical insertion [Haussler 1983]

\[ x \leftarrow y = \{x_1yx_2 \mid x = x_1x_2\}. \]

Contextual insertion [Galiukschov 1981]

\[ x \leftarrow^C y = \{x_1uyvx_2 \mid (u, v) \in C, x = x_1uvx_2\} \]

Overlap assembly [Csuhaj-Varjú et al. 2007]

\[ x \vartriangleleft y = \{uvw \in \Sigma^+ \mid x = uv, y = vw, v \neq \varepsilon\} \]
The **outfix guided insertion** of a string $y$ into $x$ is defined as

$$x \leftarrow y = \{ x_1uzzvx_2 \mid x = x_1uvx_2, y = uzv, u, v \neq \varepsilon \}.$$ 

We say that the nonempty substrings $u$ and $v$ are **matched parts**. The matched parts form a non-trivial outfix of $y$. 
The **outfix guided insertion** of a string \( y \) into \( x \) is defined as

\[
x \leftarrow y = \left\{ x_1 uzvx_2 \mid x = x_1 uvx_2, y = uzv, u, v \neq \varepsilon \right\}.
\]

We say that the nonempty substrings \( u \) and \( v \) are **matched parts**. The matched parts form a non-trivial outfix of \( y \). We can extend this operation for languages by setting

\[
L_1 \leftarrow L_2 = \bigcup_{x \in L_1, y \in L_2} x \leftarrow y.
\]
Outfix-guided insertion is not associative.

\[ acd \leftarrow abc \leftarrow abcd \]
For a language $L$, define

- $\text{OGI}^{(0)}(L) = L$,
- $\text{OGI}^{(i+1)}(L) = \text{OGI}^{(i)}(L) \cup \text{OGI}^{(i)}(L)$,

The **outfix-guided insertion closure** of $L$ is

$$\text{OGI}^*(L) = \bigcup_{i=0}^{\infty} \text{OGI}^{(i)}(L).$$
Note that by selecting the entire string $x$ as an outfix, we have $x \in x \leftarrow x$ for all $x \in \Sigma^*$ with $|x| \geq 2$. 
Note that by selecting the entire string $x$ as an outfix, we have $x \in x \leftarrow x$ for all $x \in \Sigma^*$ with $|x| \geq 2$. This implies that for any language $L$,

$$L \setminus (\Sigma \cup \{\varepsilon\}) \subseteq \mathcal{OGI}^{(1)}(L)$$

and thus, $\mathcal{OGI}^{(i)}(L) \subseteq \mathcal{OGI}^{(i+1)}(L)$ for all $i \geq 1$. 
Let $L_1$ and $L_2$ be languages. The right one-sided iterated insertion of $L_2$ into $L_1$ is defined by setting

- $\text{ROGI}^{(0)}(L_1, L_2) = L_2$,
- $\text{ROGI}^{(i+1)}(L_1, L_2) = L_1 \leftarrow \text{ROGI}^{(i)}(L_1, L_2)$.

The right one-sided insertion closure of $L_2$ into $L_1$ is

$$\text{ROGI}^*(L_1, L_2) = \bigcup_{i=0}^{\infty} \text{ROGI}^{(i)}(L_1, L_2).$$
Let $L_1$ and $L_2$ be languages. The left one-sided iterated insertion of $L_2$ into $L_1$ is defined by setting

- $\text{LOGI}^{(0)}(L_1, L_2) = L_1$,
- $\text{LOGI}^{(i+1)}(L_1, L_2) = \text{LOGI}^{(i)}(L_1, L_2) \leftarrow L_2$.

The left one-sided insertion closure of $L_2$ into $L_1$ is

$$\text{LOGI}^*(L_1, L_2) = \bigcup_{i=0}^{\infty} \text{LOGI}^{(i)}(L_1, L_2).$$
Let \( L_1 = \{ aacc \} \), \( L_2 = \{ abc \} \).
Let $L_1 = \{ aacc \}, L_2 = \{ abc \}$.

\[ \text{ROGI}^*(L_1, L_2) = a^+ bc^+ \]
Let \( L_1 = \{ aacc \} \), \( L_2 = \{ abc \} \).

\[
\text{ROGI}^*(L_1, L_2) = a^+ bc^+
\]

\[
\text{LOGI}^*(L_1, L_2) = \{ aabcc, aacc \}
\]
Proposition

If $L_1$ and $L_2$ are regular, then so is $L_1 \leftarrow L_2$. 
Proposition

If $L_1$ and $L_2$ are regular, then so is $L_1 \leftarrow L_2$.

Construct an NFA with state set

$$Q \times (P \cup \overline{P} \cup \{\spadesuit, \heartsuit\}) \cup \overline{Q} \times P.$$
Proposition

If $L_1$ and $L_2$ are regular, then so is $L_1 \leftarrow L_2$.

Construct an NFA with state set

$$Q \times (P \cup \overline{P} \cup \{\spadesuit, \heartsuit\}) \cup \overline{Q} \times P.$$
Theorem
There exists a finite language $L$ such that $\Omega \Omega \Omega^*(L)$ is nonregular.
Theorem

There exists a finite language $L$ such that $\mathcal{O}\mathcal{G}\mathcal{I}(L)$ is nonregular.

$$L = \{a_3a_1b_1b_3, a_3a_1a_2b_1, a_2b_2b_1b_3,$$
$$a_1a_2a_3b_2, a_3b_3b_2b_1, a_2a_3a_1b_3, a_1b_1b_3b_2\}.$$
\[ L = \{ a_3 a_1 a_2 b_1, \ a_2 b_2 b_1 b_3, \ a_1 a_2 a_3 b_2, \]
\[ a_3 b_3 b_2 b_1, \ a_2 a_3 a_1 b_3, \ a_1 b_1 b_3 b_2 \} \]

\[
\downarrow
\]

\[ \$a_3 a_1 b_1 b_3\$ \]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \\
    a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \downarrow \]

\[ a_3 a_1 b_1 b_3 \]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \Downarrow \]

\[ \$ a_3 a_1 a_2 b_1 b_3 \$ \]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \Downarrow \]

\$ a_3 a_1 a_2 b_1 b_3 \$
\[
L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \}
\]

\[\Downarrow\]

\[a_3 a_1 a_2 b_2 b_1 b_3\]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \\
    a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \\
    a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \downarrow \]

\[ \$a_3 a_1 a_2 a_3 b_2 b_1 b_3\$ \]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \\
a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \downarrow \]

\[ \$ a_3 a_1 a_2 a_3 b_2 b_1 b_3 \$ \]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \\
    a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \Downarrow \]

\[ \$_{a_3 a_1 a_2 a_3 b_3 b_2 b_1 b_3}_\$ \]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \]
\[ a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \downarrow \]

\[ \{ a_3 a_1 a_2 a_3 b_3 b_2 b_1 b_3 \} \]
$$L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \}$$

\[ \Downarrow \]

\$$a_3 a_1 a_2 a_3 a_1 b_3 b_2 b_1 b_3 \$$
\[
L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \\
a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \}
\]

\[
\downarrow
\]

\[
\$ a_3 a_1 a_2 a_3 a_1 b_3 b_2 b_1 b_3 \$
\]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \\
    a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \downarrow \]

\[ \$ a_3 a_1 a_2 a_3 a_1 b_1 b_3 b_2 b_1 b_3 \$ \]
\[ L = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, \]
\[ a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \} \]

\[ \downarrow \]

\[ a_3 a_1 a_2 a_3 a_1 b_1 b_3 b_2 b_1 b_3 a_3 a_1 a_2 a_3 a_1 b_1 b_3 b_2 b_1 b_3 \]
$\mathcal{OGI}^*(L) = \{a_3(a_1a_2a_3)^i z(b_3b_2b_1)^i b_3 \mid i \geq 0, z \in S\}$

$$S = \{a_1b_1, a_1a_2b_1, a_1a_2b_2b_a, a_1a_2a_3b_2b_1, a_1a_2a_3b_3b_2b_1, a_1a_2b_1a_3b_3b_2b_1\}$$
Theorem
The outfix-guided insertion closure of a unary regular language is always regular.
Theorem
The outfix-guided insertion closure of a unary regular language is always regular.

The 2-overlap catenation of $x$ and $y$, denoted $x \odot^2 y$ is defined as the set

$$\{ z \in \Sigma^+ | (\exists u, w \in \Sigma^*)(\exists v \in \Sigma^{\geq 2}) x = uv, y = vw, z = uvw \}.$$
Theorem
The outfix-guided insertion closure of a unary regular language is always regular.

The **2-overlap catenation** of $x$ and $y$, denoted $x \circledcirc^2 y$ is defined as the set

$$\{ z \in \Sigma^+ \mid (\exists u, w \in \Sigma^*)(\exists v \in \Sigma^{\geq 2})x = uv, y = vw, z = uvw \}.$$ 

- If $x, y \in a^*$, then $x \leftarrow y = x \circledcirc^2 y$.
- If $L$ is a unary language, then $\text{OGI}^*(L) = 2\text{OC}^*(L)$.
- The 2-overlap catenation closure of a regular language is regular.
Proposition

There exist finite languages $L_1, L_2, L_3, L_4$ such that $\text{ROGI}^*(L_1, L_2)$ and $\text{LOGI}^*(L_3, L_4)$ are non-regular.
Proposition

There exist finite languages $L_1, L_2, L_3, L_4$ such that $\text{ROGI}^*(L_1, L_2)$ and $\text{LOGI}^*(L_3, L_4)$ are non-regular.

For $L_1 = \{acdb, cabd\}$ and $L_2 = \{a$\$b\}$, we have

$$\text{ROGI}^*(L_1, L_2) = \{(ca)^i$(bd)^i | i \geq 0\} \cup \{a(ca)^i$(bd)^ib | i \geq 0\}$$
Proposition

There exist finite languages $L_1, L_2, L_3, L_4$ such that $\text{ROGI}^*(L_1, L_2)$ and $\text{LOGI}^*(L_3, L_4)$ are non-regular. For $L_1 = \{ acdb, cabd \}$ and $L_2 = \{ a$b \}$, we have

$$\text{ROGI}^*(L_1, L_2) = \{(ca)^i$(bd)^i \mid i \geq 0\} \cup \{ a(ca)^i$(bd)^i b \mid i \geq 0\}$$

For $L_3 = \{ a_3 a_1 b_1 b_3 \}$ and $L_4 = \{ a_3 a_1 a_2 b_1, a_2 b_2 b_1 b_3, a_1 a_2 a_3 b_2, a_3 b_3 b_2 b_1, a_2 a_3 a_1 b_3, a_1 b_1 b_3 b_2 \}$, we have the same language as in the regular language case.
Theorem
There exists a context-free language $L$ such that $L \leftarrow L$ is not context-free.
Theorem
There exists a context-free language $L$ such that $L \leftarrow L$ is not context-free.

\[
L = \{a^n c^n \mid n \geq 1\} \cup \{a^n b^n \mid n \geq 1\}
\]
Theorem
There exists a context-free language $L$ such that $L \leftarrow L$ is not context-free.

$L = \{a^n c^n \mid n \geq 1\} \cup \{a^n b^n \mid n \geq 1\}$

$(L \leftarrow L) \cap a^+ b^+ c^+ = \{a^n b^n c^n \mid n \geq 1\}$
Theorem
If $L_1$ is context-free and $L_2$ is regular, then $L_1 \leftarrow L_2$ and $L_2 \leftarrow L_1$ are context-free.

The same idea as for the case of regular $L_1$ and $L_2$ with the addition of stack operations for the context-free language.
Theorem
If $L_1$ is deterministic context-free and $L_2$ is regular, then $L_1 \leftarrow L_2$ and $L_2 \leftarrow L_1$ need not be deterministic context-free.
Theorem
If $L_1$ is deterministic context-free and $L_2$ is regular, then $L_1 \leftarrow L_2$ and $L_2 \leftarrow L_1$ need not be deterministic context-free.

For $L_1 = \{cda^ib^ia^j | i, j \geq 1\} \cup \{ca^ib^ja^j | i, j \geq 1\}$ and $L_2 = \{cda\}$, we have

$$L_1 \leftarrow L_2 = cd \cdot (\{a^ib^ja^j | i, j \geq 1\} \cup \{a^ib^ja^j | i, j \geq 1\}).$$
Theorem
If $L_1$ is deterministic context-free and $L_2$ is regular, then $L_1 \leftarrow L_2$ and $L_2 \leftarrow L_1$ need not be deterministic context-free.

For $L_1 = \{cd a^i b^i a^j | i, j \geq 1\} \cup \{ca^i b^i a^j | i, j \geq 1\}$ and $L_2 = \{cd a\}$, we have

$$L_1 \leftarrow L_2 = cd \cdot (\{a^i b^i a^j | i, j \geq 1\} \cup \{a^i b^i a^j | i, j \geq 1\}).$$

For $L_3 = (a^* bac) + (aba^*)$ and $L_4 = \{b^i a^j c | j \geq 1\} \cup \{a^i b^i a^2 | i \geq 1\}$, we have

$$L_3 \leftarrow L_4 = \{a^i b^i a^j c | i, j \geq 1\} \cup \{a^i b^i a^j | i \geq 1, j \geq 2\}.$$
We say that a language $L$ is closed under outfix-guided insertion if outfix-guided insertion of strings of $L$ into $L$ does not produce strings outside of $L$. That is, $(L \leftarrow L) \subseteq L$. 
Proposition

There is a polynomial time algorithm to decide whether for a given DFA $A$ the language $L(A)$ is og-closed.
Proposition

There is a polynomial time algorithm to decide whether for a given DFA $A$ the language $L(A)$ is og-closed.

- Construct NFA $B$ for $L(A) \leftarrow L(A)$.
- Let $A'$ be the DFA obtained from $A$ by interchanging final and non-final states.
- $L(B) \subseteq L(A)$ if and only if $L(B) \cap L(A') = \emptyset$. 
Theorem
For a given context-free language $L$, the question of whether or not $L$ is og-closed is undecidable.

- Via a PCP instance.
- Outfix-guided insertion of two regular languages is regular.
- There exist outfix-guided closures of finite languages that are non-regular.
- Outfix-guided insertion of two context-free languages may be non-context-free.
- Outfix-guided insertion of a context-free language and regular language is context-free.
- Outfix-guided insertion of a deterministic context-free language and regular language is not deterministic context-free.
- Deciding outfix-guided closure for a regular language is decidable and can be computed in polynomial time if given as a DFA.
Some open problems:

- Does there exist a regular language $L$ such that the outfix-guided insertion closure of $L$ is not context-free?
- If $L$ is context-free, is $\text{OGI}^*(L)$ context-sensitive?
- What is the complexity of deciding outfix-guided closure for a language given an NFA?