Simple gene assembly is deterministic

Presentation 10.03.2006

Miika Langille
Turku Centre for Computer Science
Department of Computer Science, Åbo Akademi University
Turku 20520 Finland
miika.langille@abo.fi

Ion Petre
Academy of Finland and
Turku Centre for Computer Science
Department of Computer Science, Åbo Akademi University
Turku 20520 Finland
ion.petre@abo.fi

Introduction: Previous research

- Simple operations were first presented in: Modelling simple operations for gene assembly, by T. Harju, I. Petre, and G. Rozenberg
- This was a restriction of the general model.
- Formalised for MDS descriptors, signed permutations and signed strings.

Introduction: Previous research

- Simple operations were initially defined in such a way, that only a single MDS (or a single composite MDS) could be moved or inverted in one operation.
  - Not universal.
    - Let $\mathcal{M}_n = \overline{\mathcal{M}}_1 \, \mathcal{M}_4 \, \mathcal{M}_3 \, \mathcal{M}_2$. No simple operation can be called on this MDS descriptor.
  - Nondeterministic in the sense that they could sort to different permutations.
- They were only investigated in-depth in their elementary form.
  - Found to be nondeterministic in that there were permutations with successful and unsuccessful strategies.

Goal

- We set out to investigate the sorting power and characteristics of the unrestricted simple operations. This involved:
  - Reconstructing the definitions,
  - Finding characteristics of simple permutations,
  - How to decide if a given permutation is sortable.

New model for simple operations

- Uses signed permutations.
- Redefined operations to fully accommodate composites.
  - After an operation has been performed, the newly formed composite is considered to be a single MDS, thus allowing further operations to be called.
- Standardised naming of operations.
  - An operation is always named after the smallest pointer involved in that operation.

Genes as signed permutations

- Represent each MDS as the integer indicating its position in the macronuclear gene.
  - The alphabet becomes $\Sigma_n = \{1, 2, \ldots, n\}$.
- A micronuclear or intermediate gene is now a signed permutation of the orthodox permutation.
- In signed permutations, pointers are never removed.
  - Consecutive adjacent pointers are considered to be a composite MDS. e.g. 345 and $\overline{543}$ both represent composites of length 3.
  - Id is assumed to occur as soon as letters are placed next to each other.

Gene assembly as a sorting of signed permutations

Gene assembly has now become the process of sorting a signed permutation, with a *sorted* permutation defined as follows.

Definition 1. We say that a signed permutation μ is *sorted* if either:

i. $\mu = i(i+1)\dots n1\dots(i-1)$, or

ii. $\mu = \overline{(i-1)}\dots\overline{1}\overline{n}\dots\overline{(i+1)}\overline{i}$,

for some $1 \leq i \leq n$. If $i = 1$ we say that μ is a *linear* permutation. We call μ *circular* otherwise. In case i. we say that μ is sorted in the *orthodox order*, while in case ii. we say that μ is sorted in the *inverted order*.

Simple Hi

Only one IES may separate the composite containing the pointer p and the inverted composite which follows it in the macronuclear gene.

For each $p \geq 1$, sh_p is defined as follows:

$$\mathsf{sh}_p(xp \dots (p+i)(\overline{p+k}) \dots (\overline{p+i+1})y) = xp \dots (p+i)(p+i+1) \dots (p+k)y,$$
$$\mathsf{sh}_p(x(\overline{p+i}) \dots \overline{p}(p+i+1) \dots (p+k)y) = xp \dots (p+i)(p+i+1) \dots (p+k)y,$$
$$\mathsf{sh}_p(x(p+i+1) \dots (p+k)(\overline{p+i}) \dots \overline{p}) = x(\overline{p+k}) \dots (\overline{p+i+1})(\overline{p+i}) \dots \overline{p}y,$$
$$\mathsf{sh}_p(x(\overline{p+k}) \dots (\overline{p+i+1})p \dots (p+i)y) = x(\overline{p+k}) \dots (\overline{p+i+1})(\overline{p+i}) \dots \overline{p}y,$$

where $k > i \geq 0$ and x, y, z are signed strings over Σ_n. We denote $\mathsf{SH} = \{\mathsf{sh}_i \mid 1 \leq i \leq n\}$.

Simple Dlad

The pointer preceding and the pointer following the composite begun with the pointer p must be separated by only one IES.

For each p, $2 \leq p \leq n-1$, sd_p is defined as follows:

$$\mathsf{sd}_p(x\,p\ldots(p+i)\,y\,(p-1)\,(p+i+1)\,z) = xy(p-1)p\ldots(p+i)(p+i+1)z,$$
$$\mathsf{sd}_p(x\,(p-1)(p+i+1)yp\ldots(p+i)z) = x(p-1)p\ldots(p+i)(p+i+1)yz,$$

where $i \geq 0$ and x, y, z are signed strings over Σ_n. We also define $\mathsf{sd}_{\overline{p}}$ as follows:

$$\mathsf{sd}_{\overline{p}}(x\overline{(p+i+1)}\overline{(p-1)}y\overline{(p+i)}\ldots\overline{p}z) = x\,\overline{(p+i+1)}\overline{(p+i)}\ldots\overline{p}\overline{(p-1)}yz,$$
$$\mathsf{sd}_{\overline{p}}(x\overline{(p+i)}\ldots\overline{p}y\overline{(p+i+1)}\overline{(p-1)}z) = xy\overline{(p+i+1)}\overline{(p+i)}\ldots\overline{p}\overline{(p-1)}z,$$

where $i \geq 0$ and x, y, z are signed strings over Σ_n. We denote
$\mathsf{SD} = \{\mathsf{sd}_i, \mathsf{sd}_{\overline{i}} \mid 1 \leq i \leq n\}$.

Characteristics of simple operations

It was stated in the original definition of simple operations that they were nondeterministic. This was due to an error in one of the examples, although it was shown to be true for the elementary operations.

To be able to explain our findings, regarding determinism, we must first define what we termed the *structure* of a permutation.

Structure of a permutation

- Relevant characteristics of a permutation.
- First and last letters of a composite MDS.
- Those in the middle play no further part in gene assembly.

For any $i \in \Sigma_n$ let $\xi_i : \Sigma_n^{\maltese} \to \Sigma_n^{\maltese}$ be the morphism defined by $\xi_i(i) = \lambda$, $\xi_i(k) = k$, for $1 \leq k < i$, and $\xi_i(k) = k - 1$, for $i < k \leq n$.

Consider also the mapping $\sigma_i : \Sigma_n^{\maltese} \to \Sigma_n^{\maltese}$ where $\sigma_i(u)$, $u \in \Sigma_n^{\maltese}$, is defined as follows:

i. If $i(i+1) \not\leq u$ and $\overline{(i+1)}\bar{i} \not\leq u$, then $\sigma_i(u) = u$ and

ii. Otherwise, $\sigma_i(u) = \xi_i(u)$.

Definition 2. We say that a word $u \in \Sigma_n^{\maltese}$ is *simple* if $\sigma_i(u) = u$, for any $i \in \Sigma_n$. Equivalently, $i(i+1) \not\leq u$ and $\overline{(i+1)}\bar{i} \not\leq u$, for any $i \in \Sigma_n$. We say that u is a *simple word associated to* $v \in \Sigma_n^{\maltese}$ if $u = \sigma_{i1}\sigma_{i2}\ldots\sigma_{ik}(v)$, for some $i_1, i_2, \ldots, i_k \in \Sigma_n$.

Operations on simple permutations

- An operation is still applicable even after suppressing the permutation to a simple permutation.
- The result of an operation applied to a permutation and to its simple equivalent both have the same structure.

Example 1. i. The simple permutation associated to $\pi_1 = 18567\overline{432}$ is $\sigma(\pi_1) = 143\overline{2}$.

ii. The simple permutation associated to $\pi_2 = 1234$ is $\sigma(\pi_2) = 1$.

iii. The simple permutation associated to $\pi_3 = 3412$ is $\sigma(\pi_3) = 21$.

iv. Operation sh_3 is applicable to $\pi_4 = 12\overline{3}4$ and $\mathsf{sh}_3(\pi_4) = 1234$. In this case, $\sigma(\pi_4) = 1\overline{2}3$, $\sigma(3) = 2$ and $\sigma(\mathsf{sh}_3) = \mathsf{sh}_2$. Note that $\sigma(\mathsf{sh}_3)$ is applicable to $\sigma(\pi_4)$; indeed, $\sigma(\mathsf{sh}_3)(\sigma(\pi_4)) = \sigma(\mathsf{sh}_3(\pi_4))$.

Operations on simple permutations

The operations applicable to simple permutations have in fact become the same ones defined for elementary operations.

For a simple permutation π, if sh_p is applicable to π, $p \in \Sigma_n$, then

$$\pi \in \{xp(\overline{p+1})y,\ x(\overline{p+1})py,\ x(p+1)\overline{p}y,\ x\overline{p}(p+1)y \mid x, y \in \Sigma_n^{\maltese}\}, \tag{1}$$

Also, if sd_p is applicable to π, $p \in \Sigma_n$, then

$$\pi \in \{x(p-1)(p+1)ypz,\ xpy(p-1)(p+1)z \mid x, y, z \in \Sigma_n^{\maltese}\}, \tag{2}$$

Similarly, if $\mathsf{sd}_{\overline{p}}$ is applicable to π, then

$$\pi \in \{x\overline{p}y(\overline{p+1})(\overline{p-1})z,\ x(\overline{p+1})(\overline{p-1})y\overline{p}z \mid x, y, z \in \Sigma_n^{\maltese}\}. \tag{3}$$

Confluent strategies

- Sorting strategies for a given signed permutation are confluent:
  - All strategies are successful, or
  - All strategies are unsuccessful.
- Nondeterminism generally occurs when there is choice.
- To prove our claim, we need to consider the interactions of two (or more) operations which can be applied to a signed permutation.

Simple operation interaction: $\mathsf{sh} + \mathsf{sh}$

- Can two sh operations affect each other?
  - Yes, but in a very limited way.

Lemma 1. *Let π be a simple signed permutation over Σ_n and $\phi, \psi \in \mathsf{SH}$ be two operations applicable to π. Then either $\phi(\pi) = \psi(\pi)$, or $\phi(\psi(\pi)) = \psi(\phi(\pi))$.*

Proof. Let $\phi = \mathsf{sh}_p$ and $\psi = \mathsf{sh}_q$, for some $p \neq q$. It follows from (1) that if $|p - q| > 1$, then $\phi(\psi(u)) = \psi(\phi(u))$.

Now, if $|p - q| = 1$, say $q = p + 1$, then

$$\pi \in \{x\overline{p}(p+1)(\overline{p+2})y,\ x(\overline{p+2})(p+1)\overline{p}y,\ xp(\overline{p+1})(p+2)y,\ x(p+2)(\overline{p+1})py\},$$

for some x, y signed words over Σ_n. Clearly, in this case, $\phi(\pi) = \psi(\pi)$. □

Simple operation interaction: $\mathsf{sd} + \mathsf{sd}$

- Can two sd operations affect each other?
  - Yes, but with interesting results.

Lemma 2. *Let π be a simple signed permutation over Σ_n and $\phi, \psi \in \mathsf{SD}$ be two operations applicable to π. Then either $\phi(\psi(\pi)) = \psi(\phi(\pi))$, or $\phi(\pi)$ and $\psi(\pi)$ have the same structure.*

Proof. Let $\phi = \mathsf{sd}_p$ and $\psi = \mathsf{sd}_q$, for some $p \neq q$. Clearly, if one of them is signed and the other is unsigned, then $\phi(\psi(\pi)) = \psi(\phi(\pi))$. Assume then that $p, q \in \Sigma_n$. The case $p, q \in \overline{\Sigma}_n$ is completely similar.

It follows from (2) that if $|p - q| > 1$, then $\phi(\psi(\pi)) = \psi(\phi(\pi))$. If $|p - q| = 1$, say $q = p + 1$, it follows from (2) that $\pi = x(p-1)(p+1)yp(p+2)z$ or $\pi = xp(p+2)y(p-1)(p+1)z$, for some x, y, z. In the former case, $\phi(\pi) = x(p-1)p(p+1)y(p+2)z$ and $\psi(\pi) = x(p-1)yp(p+1)(p+2)z$ have the same structure. The latter case is completely similar. □

Simple operation interaction

An sh and an sd operation cannot affect each other because of their definitions. sh requires a positive pointer and in sd they must all be negative.

This can now be easily extended to all signed permutations.

Example 2. Let $\pi = 135246$.

i. Both sd_2 and sd_5 are applicable to π. Moreover, $\mathsf{sd}_2 \circ \mathsf{sd}_5$ and $\mathsf{sd}_5 \circ \mathsf{sd}_2$ are also applicable to π and $\mathsf{sd}_2 \circ \mathsf{sd}_5(\pi) = \mathsf{sd}_5 \circ \mathsf{sd}_2(\pi) = 123456$.

ii. Both sd_2 and sd_3 are applicable to π. Moreover, $\mathsf{sd}_2(\pi) = 123546$ and $\mathsf{sd}_3(\pi) = 152346$ have the same structure: $\sigma(\mathsf{sd}_2(\pi)) = \sigma(\mathsf{sd}_3(\pi)) = 1324$.

Confluent strategies

- All strategies for any given signed permutations are either:
  - All successful, or
  - All unsuccessful.
- Moreover, all strategies lead to the same simple permutation.
  - Even though the full permutation may be different.
- This implies that simple operations are indeed deterministic in a weak sense.

This is an interesting result because it implies that the ciliate could just apply the operations in the order that it finds them, with no concern for incorrect assembly.

Quadratic decidability

Since gene assembly with simple operations is deterministic, deciding whether a permutation is sortable can be done by just applying the first available operation, and repeating until there are no more valid operations. If the result is sorted, then the permutation was sortable.

This gives us a method for deciding sortability in quadratic time.

Open problems

Some open problems that we have not yet fully investigated include:

- Linear method for deciding sortability.
- Structural characteristics of sortable or unsortable permutations.
- Sortable permutations of length n.