B. FILTERING MODELS

B.4 Parallel filtering model

1. The parallel filtering model was developed in the mid 90s by Martyn Amos and colleagues to be a means of describing DNA algorithms for any NP problem (as opposed to Ableson and Lipton, which are specialized to particular problems).

2. “Our choice is determined by what we know can be effectively implemented by very precise and complete chemical reactions within the DNA implementation.”

3. All PFM algorithms begin with a multi-set of all candidate solutions.

4. Mark and destroy: The PFM differs from others in that removed strings are discarded and cannot be used in further operations. Therefore is is a “mark and destroy” approach to DNA computation.

B.4.a Basic operations

1. The basic operations are remove, union, copy, and select.

2. Remove: The operation remove($U, \{S_1, \ldots, S_n\}$) removes from $U$ any strings that contain any of the substrings $S_i$.

3. Remove is implemented by two primitive operations, mark and destroy.

4. mark: mark($U, S$) marks all strands that have $S$ as a substring. This is done by adding $\overline{S}$ as a primer with polymerase to make it double-stranded.

5. destroy: destroy($U$) removes all the marked sequences from $U$. This is done by adding a restriction enzyme that cuts up the double-stranded part. These fragments can be removed by gel electrophoresis, or left in the solution (since they won’t affect it).

6. Restriction enzymes are much more reliable than other DNA operations, which is one advantage of the PFM approach.

---

6Amos, p. 50.
5.6 Implementation of the Parallel Filtering Model

Here we describe how the set operations within the Parallel Filtering Model described in Section 3.2 may be implemented.

- **Remove** \((U, \{S_i\})\) implemented as a composite operation, comprised of the following:
  - **mark** \((U, S)\) implemented by adding to \(U\) many copies of a primer corresponding to \(S\) (Fig. 5.7b). This primer only anneals to single strands containing \(S\). When DNA polymerase extends the primers once they have annealed, making only the single strands containing \(S\) double stranded (Fig. 5.7b).
  - We may then **destroy** strands containing \(S\) by adding the appropriate restriction enzyme. Double-stranded DNA (i.e. strands marked as containing \(S\)) is cut at the restriction sites embedded within, single strands remaining intact.

**Figure IV.8:** *Remove* operation implemented by *mark* and *destroy*. [source: Amos]

\[\begin{align*}
\text{(a)} & \quad \text{Restriction site} & \text{Target sequence} \\
\text{(b)} & \quad \text{Polymerase extends} & \text{Primer block} \\
\text{(c)} & \quad \text{Restrict} & \text{Restrict} & \text{Restrict} \\
\text{(d)} & \quad \text{Restrict} & \text{Restrict} & \text{Restrict}
\end{align*}\]

\[\begin{align*}
\text{¶7. Union:} & \quad \text{The operation } \text{union}(\{U_1, \ldots, U_n\}, U) \text{ combines in parallel the multi-sets } U_1, \ldots, U_n \text{ into } U. \\
\text{¶8. Copy:} & \quad \text{The operation } \text{copy}(U, \{U_1, \ldots, U_n\}) \text{ divides multi-set } U \text{ into } n \text{ equal multi-sets } U_1, \ldots, U_n. \\
\text{¶9. Select:} & \quad \text{The operation } \text{select}(U) \text{ returns a random element of } U. \text{ If } U = \emptyset, \text{ then it returns } \emptyset. \\
\text{¶10. Homogeneous DNA can be detected and sequenced by PCR.} \\
& \quad \text{Nested PCR can be used in non-homogeneous cases (multiple solutions).} \\
\text{¶11. These operations are assumed to be constant-time.} \\
\text{¶12. Periodic amplification (especially after copy operations) may be necessary to ensure an adequate number of instances.} \\
\text{¶13. Amos et al. have done a number of experiments to determine optimum reactions parameters and procedures.}
\end{align*}\]
B. FILTERING MODELS

B.4.b Permutations

1. **Input:** “The input set $U$ consists of all strings of the form $p_1i_1p_2i_2\cdots p_ni_n$ where, for all $j$, $p_j$ uniquely encodes ‘position $j$’ and each $i_j$ is in $\{1, 2, \ldots, n\}$. Thus each string consists of $n$ integers with (possibly) many occurrences of the same integer.”

2. **for** $j = 1$ **to** $n - 1$ **do**
   
   copy($U_1, U_2, \ldots, U_n$)
   
   **for** $i = 1, 2, \ldots, n$ and all $k > j$
   
   **in parallel** **do** remove($U_i, \{p_{ji} \neq p_{ki} \}$)
   
   // $U_i$ contains $i$ in $j$th position and no other $i$
   
   union($\{U_1, U_2, \ldots, U_n\}, U$)
   
   **end for**
   
   $P_n \leftarrow U$

3. In the preceding, remove($U_i, \{p_{ji} \neq p_{ki} \}$) means to remove from $U_i$ all strings that have a $p_j$ value not equal to $i$ and all strings containing $p_ki$ for any $k > j$.

   For example, if $i = 2$ and $j = n - 1$, this remove operation translates to remove($U_2, \{p_{n-1}1, p_{n-1}3, p_{n-1}4, \ldots, p_{n-1}n, p_{n}2\}$).

   That is, it eliminates all strings except those with 2 in the $n-1$ position, and eliminates those with 2 in the $n$ position.

4. At the end of iteration $j$ we have:

   $$\alpha_{p_1i_1p_2i_2\cdots p_{ji}p_{j+1i_{j+1}}\cdots p_ni_n} \beta$$

   where $\alpha$ represents a permutation of $j$ integers from 1, $\ldots$, $n$, and none of these integers $i_1, \ldots, i_j$ are in $\beta$.

5. **NP-complete problems:** Amos shows how to do a number of NP-complete problems, including 3-vertex-colorability, HPP, subgraph isomorphism, and maximum clique.

---

7Amos, p. 51.