

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.”⁶
- ¶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 $\text{remove}(U, \{S_1, \dots, S_n\})$ 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:** $\text{mark}(U, S)$ marks all strands that have S as a substring. This is done by adding \bar{S} as a primer with polymerase to make it double-stranded.
- ¶5. **destroy:** $\text{destroy}(U)$ 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.

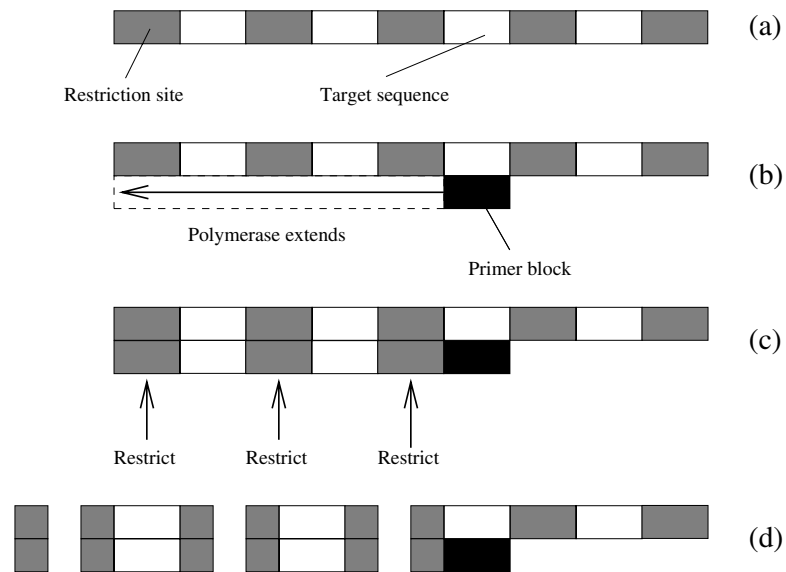


Figure IV.8: *Remove* operation implemented by *mark* and *destroy*. [source: Amos]

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

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 $\{1, 2, \dots, n\}$. Thus each string consists of n integers with (possibly) many occurrences of the same integer.”⁷

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

¶3. In the preceding, remove($U_i, \{p_ji_j \neq p_ji, p_ki\}$) 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($U_2, \{p_{n-1}1, p_{n-1}3, p_{n-1}4, \dots, p_{n-1}n, p_n2\}$).
 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:

$$\overbrace{p_1i_1p_2i_2\cdots p_ji_j}^{\alpha} \underbrace{p_{j+1}i_{j+1}\cdots p_ni_n}_{\beta}$$

where α represents a permutation of j integers from $1, \dots, n$, and none of these integers i_1, \dots, i_j are in β .

¶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.