Minimal Set Cover Problem: On a DNA Solution of Selection Stage

Pérez-Jiménez, M.J.; Sancho-Caparrini, F.
Dpt. Computer Science and Artificial Intelligence. Universidad de Sevilla
E-mail: {marper,fsancho}@cica.es

Abstract. The introduction of Sticker Model by S. Roweis et al. ([4]) is illustrated with a solution in this model of a **NP**-complete problem: Minimal Set Cover Problem ([1]). The molecular solution given in [4] has three stages, the last one is a subroutine to select a minimal set cover of a finite set from a collection of covers of it. In this work a formal verification of this subroutine is given through a systematic method using a labeling procedure and invariant formulas searching.

1 Introduction

Given a molecular program P designed to solve a problem X, the verification of (X, P) consists in proving that the program P solves the problem X. We consider molecular programs that start with an initial test–tube (a finite multiset of elements over a prefixed alphabet) as input. They return a (possibly empty) final tube of answers as output. In order to establish the formal verification of a molecular program designed to answer a problem, it suffices to show two basic results:

- *Soundness (of the program)*: every strand of the final test–tube encodes a correct solution of the problem.
- *Completeness (of the program)*: every strand of the initial test–tube encoding a correct solution of the problem is kept alive along the execution of the program (and it is placed in the "corresponding" output tube).

Formal verification of molecular programs is a necessary step for their treatment with an automated reasoning system.
In order to study formal verification of molecular programs we propose the structured and systematic method that follows:

1. Designing a procedure for tubes labeling in order to individualize (treat separately and in detail) the data of the model. This allows us a precise study of them through the program execution.
2. Searching invariant formulas in order to extract a set of properties that are valid along the execution. This properties will let us to establish soundness and completeness of the program.

We think that this procedure of formal verification can be adapted to membrane computing, through an *annotated specification mechanism* in some *remark points* to be considered along the P-system execution, inspired by Hoare's specification ideas ([2]).
In this work we will illustrate this method with the study of formal verification of a molecular program in the sticker model that solves Selection Problem associated to the Minimal Set Cover.

2 Minimal Set Cover Selection Problem

The selection problem associated to the Minimal Set Cover consists, basically, in sorting according to their cardinality a family of covers of a given finite set. The introduction of *sticker model* by S. Roweis et al ([4]) is illustrated by presenting in this model a solution to the minimal set cover problem ([1]). Molecular solution given in [4] is founded in a subroutine that solves the above selection problem. We study a rooted graph structure that arise through the execution of the subroutine, and then we apply the above structured method to establish formal verification of it.
The *sticker model* is an abstract model of molecular computing based on DNA that has a random access memory and it uses a new form of encoding the information. The following operations from sticker model are used in this paper:

- `Combine` (T_1, T_2): the memory complexes from the tubes T_1, T_2 are combined to form the multiset union of all strings in the two input tubes.
- `Separate` (T, i): Given a tube, T, and an integer, i ($1 \leqslant i \leqslant$ number of substrands that form each complex of T), create two new tubes, $+(T, i)$ and $-(T, i)$, where $+(T, i)$ (resp. $-(T, i)$) contain all strings of T having the i–th substrand set to 1 (resp. set to 0). We write $(T_1, T_2) \leftarrow \texttt{Separate}(T, i)$ to indicate that $T_1 = +(T, i)$ and $T_2 = -(T, i)$.

Next we formulate the above mentioned problem which will be studied in this paper.

Minimal Set Cover Selection Problem: *Given a finite set $A = \{1, ..., p\}$ and a finite family $\mathcal{F} = \{B_1, ..., B_q\}$ of subsets of A, sort the collection of all subfamilies of $\mathcal{F}$ covering A, according to their cardinality.*

A molecular program in the sticker model solving Minimal Set Cover Selection Problem is the following one:

```
Input: T_0 (encoding all sub-families of F covering A)
       For i ← 0 to q − 1 do
             T_{i+1} ← ∅
             For j ← i to 0 do
                   T_j^+ ← +(T_j, i + 1) ; T_j^− ← −(T_j, i + 1)
                   T_{j+1} ← combine(T_j^+, T_{j+1})
                   T_j ← T_j^−
```

The number of molecular operations in this program is quadratic in the size of the family $\mathcal{F}$.
For each i ($1 \leq i \leq q$) we will note $r_i = |B_i|$ and $B_i = \{x_i^1, \ldots, x_i^{r_i}\}$. If $\sigma \in T_0$, then we will write $\sigma(i) = 1$ (resp. 0) if the i-th region of memory complex σ is on (resp. off).
Note that the input tube is not a library (as usual in the sticker model), since this program is in fact a subroutine. Moreover, this program doesn't enclose operations which modify inner structure of the strands (`set`, turn on, or `clear`,

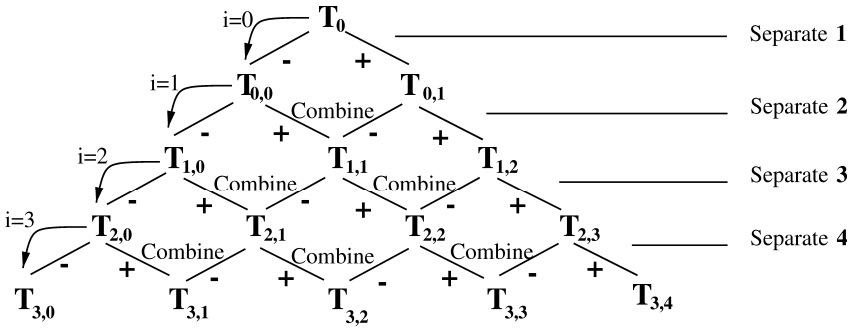
turn off). So, this subroutine can also be described in any molecular computation model without memory and based in filtering procedure.
Furthermore, if $\sigma \in T_0$, then $(\sigma(1), \ldots, \sigma(q))$ encodes in a natural way a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ as follows: $\forall\, i\ (1 \leq i \leq q \rightarrow (B_i \in \mathcal{F}' \leftrightarrow \sigma(i) = 1))$. Also, $\forall\, j\ (q+1 \leq j \leq q+p \rightarrow \sigma(j) = 1)$. We define $|\sigma| = \sum_{1<i<q} \sigma(i)$.
According to this, the input tube used in this subroutine can be unloaded in the following way: we can use a restriction endonuclease enzyme so that when encoding the memory strands of initial test tube (a $(p+q, q)$-library), an appropriate recognition site associated to the restriction enzyme is placed between q-th and $(q+1)$-th regions. In this way, just before running this subroutine, we activate the restriction endonuclease enzyme to make all memory strands split in the specific recognition site; after that, we select all molecules containing the first q regions (using magnetic beads, for example). Therefore we make that the input tube of this subroutine (noted here as T_0, too) contains all q regions length molecules encoding subfamilies of $\mathcal{F}$ covering A.

3 Formal Verification of the subroutine

To establish the formal verification of the designed subroutine, we begin with a labeling procedure of tubes that have been used along the execution.

```
Input: T_0
       T_{-1,0} ← T_0 ; T_{-1,1} ← ∅ ; T_{0,1} ← T_0 ; T*_{0,1} ← ∅
       For i ← 0 to q − 1 do
             T_{i,i+2} ← ∅
             For j ← i to 0 do
                   T+_{i,j} ← +(T_{i−1,j}, i + 1)
                   T_{i,j+1} ← combine(T+_{i,j}, T*_{i,j+1})
                   T*_{i,j} ← −(T_{i−1,j}, i + 1)
             T_{i,0} ← T*_{i,0}
```

This program returns $i+1$ tubes $(T_{i,i+1}, \ldots, T_{i,0})$ after the execution of the i-th step of the main loop.



That is, the execution of this program can be described as a rooted directed graph with depth q that we will note as *labeled merge-binary tree* with depth q, and can be defined as follows:

Definition 1. *A grid with depth h, $G_h = (V_h, E_h)$, is the following directed graph:*

$$\begin{cases} V_h = \{(i,j) :\ 0 \leq i \leq h \wedge 0 \leq j \leq i\} \\ E_h = \{((i,j),(i+1,j)),\ ((i,j),(i+1,j+1)) :\ 0 \leq i < h \wedge 0 \leq j \leq i\} \end{cases}$$

Definition 2. *A labeled merge-binary tree with depth h is a 5-tuple $(G_h, L, \{F_i : 0 \leq i \leq h-1\}, B, T_0, l)$, where:*

- *G_h is a grid with depth h.*
- *L is a nonempty set (its elements will be called labels).*
- *For each i $(0 \leq i \leq h-1)$ a function $F_i : L \longrightarrow L \times L$ (we will note $F_i = (F_i^-, F_i^+)$).*
- *A binary function $B : L \times L \longrightarrow L$.*
- *$T_0 \in L$.*
- *$l : V_h \longrightarrow L$, called the labeling function, is defined by recursion (using T_0, F_i and B), as follows:*

$$\begin{cases} l(0,0) = T_0 \\ l(i+1,0) = F_i^-(l(i,0)) \\ l(i+1,i+1) = F_i^+(l(i,i)) \\ l(i+1,j) = B(F_i^-(l(i,j-1)), F_i^+(l(i,j))) \end{cases}$$

with $0 \leq i < h \wedge 1 \leq j \leq i$.

In the execution of the designed molecular program a labeled merge-binary tree with depth q is obtained, where:

(a) Labels are tubes.
(b) For each i $(0 \leq i \leq q-1)$ we have $F_i(T) = \texttt{separate}(T, i+1)$, so
$$F_i^-(T) = -(T, i+1) \text{ and } F_i^+(T) = +(T, i+1)$$
(c) The binary function, B, is the `combine` molecular operation (called `merge` too).
(d) T_0 is the input test–tube of the subroutine.

This combinatorics structure let us design a subroutine that solves a more general sorting problem where the semantic of the subroutine is very close to the semantic of the problem ([3]).
To establish the formal verification of the subroutine, in connection with Minimal Set Cover Selection Problem, we have to prove specifically that:

- Every molecule, σ, in the output tube, $T_{q-1,r}$ $(1 \leq r \leq q)$, must verify $|\sigma| = r$ (*Soundness*).
- Every molecule, σ, in the input tube, T_0, such that $|\sigma| = r$ $(1 \leq r \leq q)$, must be in the output tube $T_{q-1,r}$ (*Completeness*).

The execution of the subroutine can be seen as an evolution of a population of elements. Initially, the population is determined by the multiset of molecules in the input tube, T_0. Every molecule is an element, and repeated ones can exist at the same time (so cloned members can be alive simultaneously in this population).
Every step of the main loop can be interpreted as a time unit. After a lapse, the population is transformed into other one, but, in this case, there is no deaths or mutations in elements, since this subroutine is, basically, a filtering procedure.
We consider the following formulas:

$$\psi(i,0) \equiv \forall\sigma\ (\sigma \in T_{i,0} \leftrightarrow \sigma \in T_0 \wedge \forall k\ (1 \leq k \leq i+1 \rightarrow k \notin \sigma))$$
$$\psi(i,j+1) \equiv \forall\sigma(\sigma \in T_{i,j+1} \leftrightarrow (\sigma \in T_{i-1,j} \wedge i+1 \in \sigma) \vee \vee\ (\sigma \in T_{i-1,j+1} \wedge i+1 \notin \sigma))$$

Theorem 1. *The formula $\theta(i) \equiv \forall j\ (0 \leq j \leq i+1 \rightarrow \psi(i,j))$ is an invariant of the main loop. That is $\forall\, i\ (0 \leq i \leq q-1 \rightarrow \theta(i))$.*

Proof. By induction on i. Base case follows from the definition of initial tubes. Let $i < q-1$ such that $\theta(i)$. It can be proved by induction on j that

$$\forall\, j\ (0 \leq j \leq i+2 \rightarrow \psi(i+1,j))$$

□

Next, we are going to describe the trace of every molecule of the input tube along the execution of the subroutine. For this, if $\sigma \in T_0$ is given, we will write $\sigma = (i_1, \ldots, i_r)$ to note that $1 \leq i_1 < \cdots < i_r \leq q$ and

$$\forall\, j\ (1 \leq j \leq r \rightarrow \sigma(i_j) = 1) \wedge \forall\, t\, \forall\, j\ (1 \leq t \leq q \wedge i_j \neq t \rightarrow \sigma(t) = 0)$$

That is, the molecule $\sigma = (i_1, \ldots, i_r) \in T_0$ encodes in a natural way the subfamily $\mathcal{F}' = \{B_{i_1}, \ldots, B_{i_r}\}$ of $\mathcal{F}$.

Proposition 1. *Let $\sigma = (i_1, \ldots, i_r) \in T_0$, where $1 \leq i_1 < i_2 < \cdots < i_r \leq q$. Then*

(1) $\forall j\ (1 \leq j \leq r \rightarrow \sigma \in T_{i_j-1,j})$.
(2) $\forall t\ (1 \leq t \leq q - i_r \rightarrow \sigma \in T_{i_r+t-1,r})$.
(3) $\sigma \in T_{q-1,r}$.

Proof.

1. By induction on j. Base case is clear for $i_1 = 1$. If $i_1 > 1$ then we can prove that $\forall\, s\ (1 \leq s < i_1 \rightarrow \sigma \in T_{s-1,0})$. From this we have $\sigma \in +(T_{i_1-2,0}, i_1)$. Since $\psi(i_1 - 1, 1)$ is true, we conclude that $\sigma \in T_{i_1-1,1}$.
   Let j $(1 \leq j < r)$ such that $\sigma \in T_{i_j-1,j}$. If $i_{j+1} = i_j + 1$ then, from $\psi(i_{j+1}-1, j+1) \equiv \psi(i_j, j+1)$ we obtain $\sigma \in T_{i_{j+1}-1,j+1}$. If $i_{j+1} > i_j+1$, then, by induction, we can prove that $\forall\, t\ (1 \leq t \leq i_{j+1} - i_j - 1 \rightarrow \sigma \in T_{i_j+t-1,j})$. From $\psi(i_{j+1}, j+1)$ we conclude that $\sigma \in T_{i_{j+1}-1,j+1}$.
2. By induction on t. Base case follows from $\psi(i_r, r)$. If $\sigma \in T_{i_r+t-1,r}$ $(1 \leq t < q - i_r)$ then, from $\psi(i_r + t, r)$, we can conclude that $\sigma \in T_{i_r+t,r}$.
3. It is clear from 2. □

Next we will prove that the generated tubes after i-th step of the main loop, $\{T_{i,0}, T_{i,1}, \ldots, T_{i,i+1}\}$, form a "partition" of the initial test tube, T_0. This confirms that the subroutine runs a filtering procedure where no strand dies along the execution.

Proposition 2. $\forall i\ (0 \leq i \leq q-1 \rightarrow T_0 = \bigcup_{0<j<i+1} T_{i,j})$.

Proof. Let us first prove by induction on i that

$$\forall\, i\ (0 \leq i \leq q-1 \rightarrow T_0 \subseteq \bigcup_{0<j<i+1} T_{i,j})$$

Base case follows from $\theta(0)$. Let $i < q-1$ such that $T_0 \subseteq \bigcup_{0\leq j\leq i+1} T_{i,j}$ and $\sigma \in T_0$. By induction hypothesis, $\exists j\ (0 \leq j \leq i+1 \wedge \sigma \in T_{i,j})$. We can prove that $\sigma \in \bigcup_{0\leq s\leq i+2} T_{i+1,s}$, distinguishing between the cases $i+2 \in \sigma$ and $i+2 \notin \sigma$.
In the same manner we can see that

$$\forall\, i\ (0 \leq i \leq q-1 \rightarrow \bigcup_{0\leq j\leq i+1} T_{i,j} \subseteq T_0)$$

□

Proposition 3. $\forall i\ (0 \leq i \leq q-1 \rightarrow \forall r\ \forall s\ (0 \leq r < s \leq i+1 \rightarrow T_{i,r} \cap T_{i,s} = \emptyset))$.

Proof. By induction on i. Base case is easy to check. Let $i < q-1$ such that $\forall\, r\ \forall\, s\ (0 \leq r < s \leq i+1 \rightarrow T_{i,r} \cap T_{i,s} = \emptyset)$. Let r, s such that $0 \leq r < s \leq i+2$. Using formula θ, the main loop invariant, and distinguishing between the cases $s = i+2$, $r = 0 \wedge 1 \leq s \leq i+1$ and $r > 0 \wedge 1 \leq s \leq i+1$, it can be showed that $T_{i+1,r} \cap T_{i+1,s} = \emptyset$.

□

Corollary 1. $T_{q-1,0} = \emptyset$.

Proof. The proof is straightforward from proposition **1**.(3) and proposition **3**.

□

Finally, we establish soundness and completeness of the designed subroutine that solves minimal set cover selection problem.

Theorem 2. (`Soundness`) *Every strand, σ, of the final test–tube, $T_{q-1,r}$ ($1 \leq r \leq q$), must verify that $|\sigma| = r$. That is, $\forall r\ (1 \leq r \leq q \rightarrow \forall \sigma \in T_{q-1,r}(|\sigma| = r))$.*

Proof. Let $\sigma \in T_{q-1,r}$. From proposition **2** is obtained that $\sigma \in T_0$. From proposition **1**.(3), we have $\sigma \in T_{q-1,|\sigma|}$. From proposition **3** we conclude that $|\sigma| = r$.

□

Theorem 3. (`Completeness`) *Every strand, σ, of the initial test–tube, T_0, such that $|\sigma| = r$, is in the output tube, $T_{q-1,r}$. That is, $\forall\, \sigma \in T_0\ (|\sigma| = r \rightarrow \sigma \in T_{q-1,r})$. Furthermore, $\forall \sigma \in T_0\ \exists! r\ (1 \leq r \leq q \wedge \sigma \in T_{q-1,r})$.*

Proof. Let $\sigma \in T_0$ such that $|\sigma| = r$. From proposition **1**.(3) we can deduce that $\sigma \in T_{q-1,r}$. In the other hand, from proposition **3** (with $i = q-1$) there exists j such that $0 \leq j \leq i+1 \wedge \sigma \in T_{q-1,j}$. From corollary **1** we obtain $j > 0$, and from proposition **3** we conclude that j is unique.

□

4 Conclusions

In this work a methodology to study formal verification of molecular programs is given. This method is applied to a subroutine of a molecular program in the sticker model designed by S. Roweis et al ([4]). The semantic of this subroutine is far from the semantic of the problem it solves. For this, soundness and completeness of subroutine are not straightforward obtained from the invariant formulas.
The formalization of verification procedures of molecular programs would allow to automate the study of properties related to this programs. We think that this is a first step to develop executable prototypes, using automated reasoning systems of molecular computation models and, in general, of unconventional models.

References

1. Garey M.R.; Johnson D.S. *Computers and intractability*, W.H. Freeman and Company, New York, 1979.
2. Hoare, C.A.R. An axiomatic basis for computer programming, *Communications of the ACM*, 12, 576–583, 1969.
3. Pérez Jiménez, M.J.; Sancho Caparrini, F. Solving Knapsack Problems in a Sticker Based Model, *Preliminary Proceedings of The Seventh International Meeting on DNA Based Computers*, 94–104, 2001.
4. Roweis, S.; Winfree, E. et al. A Sticker Based Model for DNA Computation, *J. Comp. Biol.* 5, 615–629, 1998.