Tema IV: Resolución eficiente de problemas NP-completos en modelos moleculares

El problema 3–COL

Dado un grafo no dirigido G, determinar si existe una coloración válida de G con tres colores

El problema 3-COL es presuntamente intratable

Diseñamos un programa molecular que resuelve *eficientemente* 3–COL

- Sea $G = (V, E)$ un grafo no dirigido, con $V = \{1, \dots, n\}$
- Notaremos por $p_1, \dots, p_n$ los códigos moleculares de los nodos y por c_1, c_2, c_3 los códigos moleculares de los colores.
- Para cada i, j $(1 \leq i, j \leq n)$, notaremos el par (p_i, c_j) por $p_i(c_j)$, y convendremos que $p_i(c_j)$ indica que el nodo p_i tiene asignado el color c_j.
- Sea $\{e_1, e_2, \dots, e_p\}$ el conjunto ordenado de aristas de G. Para cada i $(1 \leq i \leq p)$, e_i^1 (resp. e_i^2) es el menor (resp. mayor) de los extremos de la arista e_i.

Una solución del problema **3-COL** en el modelo restringido

Consideremos el alfabeto:

$$\Sigma = \{(p_1, x_1, p_2, x_2, ..., p_n, x_n) : \forall i \ (1 \leq i \leq n \rightarrow (x_i = c_1 \vee x_i = c_2 \vee x_i = c_3))\}$$

Los elementos de Σ se pueden identificar con moléculas de ADN del tipo $p_1 x_1 p_2 x_2 ... p_n x_n$, cada una de las cuales representan una coloración del grafo.

El tubo de entrada, T, es el siguiente multiconjunto finito:

$$T = \{a \subseteq \Sigma : a \text{ es un conjunto unitario}\}$$

Dada una molécula $\sigma = p_1 x_1 \ldots p_n x_n \in T$ y un número i $(1 \leqslant i \leqslant n)$, notaremos por $(\sigma)_i$ el color asociado al nodo codificado por p_i.

Es decir, $p_i(c_j) \equiv (\sigma)_{p_i} = c_j$.

Idea del programa molecular:

- A partir del tubo inicial T_0, que codifica todas las posibles coloraciones del grafo con tres colores, se procede como sigue.
  - Se elabora un tubo T_1 seleccionando de T_0 las coloraciones que son válidas para el subgrafo inducido por la arista e_1. Para ello:
    - ⋆ Para cada j $(1 \leqslant j \leqslant 3)$ se coloca en T_j las coloraciones que dan color j a e_1^1 y en T_j' las coloraciones de T_j que dan a e_1^1 un color distinto de j.
    - ⋆ El tubo T_1 será la unión de los tubos T_1', T_2', T_3'.
  - Se elabora un tubo T_2 seleccionando de T_1 las coloraciones que son válidas para el subgrafo inducido por las aristas e_1, e_2 (procediendo de manera análoga).
  - El proceso se reitera p veces (siendo p el número de aristas).

Estas ideas sugieren el diseño del siguiente programa molecular:

```
Entrada:  T (contiene todas las posibles coloraciones de G)
          Para i ← 1 hasta p hacer
               T_1 ← +(T, e_i^1(c_1));  T_1^* ← −(T, e_i^1(c_1))
               T_2 ← +(T_1^*, e_i^1(c_2));  T_3 ← −(T_1^*, e_i^1(c_2))
               Para j ← 1 hasta 3 hacer
                    T_j' ← −(T_j, e_i^2(c_j))
               T ← T_1' ∪ T_2'
               T ← T ∪ T_3'
          Detectar(T)
```

Entrada: T (contiene todas las posibles coloraciones de G)

Para $i \leftarrow 1$ hasta p hacer

$\quad T_1 \leftarrow +(T, e_i^1(c_1))$; $\; T_1^* \leftarrow -(T, e_i^1(c_1))$

$\quad T_2 \leftarrow +(T_1^*, e_i^1(c_2))$; $\; T_3 \leftarrow -(T_1^*, e_i^1(c_2))$

$\quad$ Para $j \leftarrow 1$ hasta 3 hacer

$\quad\quad T_j' \leftarrow -(T_j, e_i^2(c_j))$

$\quad T \leftarrow T_1' \cup T_2'$

$\quad T \leftarrow T \cup T_3'$

Detectar(T)

Verificación formal de dicho programa molecular (I)

Para establecer la verificación, se etiquetan los tubos que se obtienen a lo largo de la ejecución, reescribiendo el programa molecular como sigue:

```
Entrada:  T (en las condiciones antes citadas)
          T^0 ← T
          Para i ← 1 hasta p hacer
               T_{i,1} ← +(T^{i-1}, e_i^1(c_1));  T*_{i,1} ← −(T^{i-1}, e_i^1(c_1))
               T_{i,2} ← +(T*_{i,1}, e_i^1(2));  T_{i,3} ← −(T*_{i,1}, e_i^1(c_2))
               Para j ← 1 hasta 3 hacer
                    T'_{i,j} ← −(T_{i,j}, e_i^2(c_j))
               \overline{T}^i ← T'_{i,1} ∪ T'_{i,2}
               T^i ← \overline{T}^i ∪ T'_{i,3}
          Detectar(T^p)
```

Verificación formal de dicho programa molecular (II)

Para probar la corrección del programa se considera la fórmula

$$\theta(i) = \forall \sigma \in T^i \; \forall k \leq i \; ((\sigma)_{e_k^1} \neq (\sigma)_{e_k^2})$$

para i verificando que $1 \leqslant i \leqslant p$.

$\theta(i)$ es verdadera si todas las moléculas del tubo T^i codifican coloraciones válidas del subgrafo de G inducido por las aristas $\{e_1, ..., e_i\}$.

Se trata de probar que $\theta(i)$ es un invariante del bucle principal del programa.

Verificación formal de dicho programa molecular (III)

Lema 1: *Para cada* i, j $(1 \leq i \leq p \ \wedge \ 1 \leq j \leq 3)$ *y cada molécula* $\sigma \in T_{i,j}$ *se tiene que:* $(\sigma)_{e_i^1} = c_j$.

Lema 2: *Para cada* i, $1 \leq i < p$, *se tiene que* $T^{i+1} \subseteq T^i$).

Teorema 1: $\theta(i)$ *es un invariante del bucle principal del programa. Es decir,* $\forall i \ (1 \leq i \leq p \rightarrow \theta(i))$.

Corolario: (*Corrección* del programa molecular) *Toda molécula del tubo de salida codifica una coloración válida del grafo. Es decir,*

$$\forall \sigma \in T^p \ (\sigma \text{ codifica una coloración válida})$$

Verificación formal de dicho programa molecular (IV)

Para probar la completitud del programa se considera la fórmula:

$$\delta(i) \equiv \forall \sigma \in T_0 \forall k \leq p \; ((\sigma)_{e_k^1} \neq (\sigma)_{e_k^2} \rightarrow \sigma \in T^i)$$

$\delta(i)$ es verdadera si toda molécula del tubo inicial que codifica una coloración válida del grafo con tres colores, aparece en todos los tubos *relevantes*, T^i, que se obtienen a lo largo de la ejecución del programa.

Teorema 2: *$\delta(i)$ es un invariante del bucle principal del programa. Es decir,* $\forall \sigma \; (\sigma \in T_0 \wedge \forall k \leq p \; ((\sigma)_{e_k^1} \neq (\sigma)_{e_k^2}) \longrightarrow \forall i \; (1 \leq i \leq p \rightarrow \sigma \in T^i))$.

Corolario: (*Completitud* del programa) Toda molécula del tubo inicial que codifica una coloración válida del grafo aparece en el tubo de salida del programa. Es decir, $\forall \sigma \; (\sigma \in T_0 \wedge \forall k \leq p \; ((\sigma)_{e_k^1} \neq (\sigma)_{e_k^2} \rightarrow \sigma \in T^p))$.

Una solución del problema **3-COL** en el modelo débil

Consideremos el siguiente alfabeto: $\Sigma = \{(p_i, c_j) : 1 \leq i \leq n \wedge 1 \leq j \leq 3\}$.

El tubo de entrada del programa molecular es el siguiente

$$T = \{\sigma \in \Sigma^n : \exists x_1 ... \exists x_n (\sigma = (p_1, x_1)(p_2, x_2)...(p_n, x_n))\}$$

Si σ es una molécula de dicho tubo, entonces para cada i, $1 \leq i \leq n$, notaremos $(\sigma)_i = x_i$.

Idea del programa molecular:

- A partir del tubo de entrada se seleccionan las moléculas que codifican coloraciones válidas con tres colores del subgrafo inducido por las aristas cuyo extremo inferior es menor o igual que 1.
  - De éstas, se seleccionan las que codifican coloraciones válidas con tres colores del subgrafo inducido por las aristas cuyo extremo inferior es menor o igual que 2.
  - Y se repite el proceso n veces.

Estas ideas sugieren el diseño del siguiente programa molecular:

```
Entrada:  T (en las condiciones antes citadas)
          Para i ← 1 hasta n − 1 hacer
               copiar(T, {T_1, T_2, T_3})
               Para j ← 1 hasta 3 hacer
                    quitar(T_j, {x_i ≠ j, p_k(c_j) :  k > i ∧ {i, k} ∈ E})
               union({T_1, T_2, T_3}, T)
          Seleccion(T)
```

Verificación formal de dicho programa molecular (I)

Para establecer la verificación, se etiquetan los tubos que se obtienen a lo largo de la ejecución, reescribiendo el programa molecular como sigue:

Entrada: T^0 (en las condiciones antes citadas)
Para $i \leftarrow 1$ hasta $n-1$ hacer
copiar$(T^{i-1}, \{T_1^{i-1}, T_2^{i-1}, T_3^{i-1}\})$
Para $j \leftarrow 1$ hasta 3 hacer
$\overline{T}_j^i \leftarrow$ quitar$(T_j^{i-1}, \{x_i \neq j, p_k(c_j) : \ k > i \wedge \{i,k\} \in E\})$
union$(\{\overline{T}_1^i, \overline{T}_2^i, \overline{T}_3^i\}, T^i)$
Seleccion(T^{n-1})

Verificación formal de dicho programa molecular (II)

Del etiquetado de tubos que ha sido introducido se deduce:

(a) La sucesión de tubos relevantes, T^i que se obtiene a lo largo de la ejecución es creciente por la relación de inclusión. Es decir,

$$\forall i \ (1 \leq i \leq n-1 \rightarrow T^i \subseteq T^{i-1})$$

(b) Para cada i, j tales que $1 \leq i \leq n-1, 1 \leq j \leq 3$, y para cada molécula $\sigma \in \overline{T}_j^i$ se verifica que $(\sigma)_i = j \wedge \forall k \ (i < k \wedge \{i, k\} \in E \rightarrow (\sigma)_k \neq j)$.

Verificación formal de dicho programa molecular (III)

Para probar la corrección del programa molecular se considera la fórmula:

$$\theta(i) = \forall\sigma \in T^i \; \forall r \; \forall s \; (1 \leq r \leq i \wedge r < s \wedge \{r,s\} \in E \rightarrow (\sigma)_r \neq (\sigma)_s)$$

para cada i $(1 \leq i \leq n-1)$.

$\theta(i)$ es verdadera si cada molécula de T^i codifica una coloración del grafo que es válida para el subgrafo inducido por el conjuntos de nodos $\{1,\ldots,i\}$.

Teorema 5: *$\theta(i)$ es un invariante del bucle principal del programa. Es decir, $\forall i \; (1 \leq i \leq n-1 \rightarrow \theta(i))$.*

Corolario 6 (Corrección del programa): *Toda molécula del tubo de salida codifica una coloración válida del grafo. Es decir,*

$$\forall\sigma \in T^{n-1} \; (\sigma \text{ codifica una coloración válida})$$

Verificación formal de dicho programa molecular (IV)

A continuación, veamos que toda molécula del tubo inicial que codifique una coloración válida del grafo, aparece en todos los tubos relevantes, T^i, a lo largo de la ejecución del programa.

Teorema 7: Sea $\sigma \in T^0$ tal que

$$\forall r \; \forall s \; (1 \leq r \leq n-1 \wedge r < s \wedge \{r, s\} \in E \rightarrow (\sigma)_r \neq (\sigma)_s)$$

Entonces $\forall i \; (1 \leq i \leq n-1 \rightarrow \sigma \in T^i)$.

Corolario (Completitud del programa): *Sea $\sigma \in T^0$ tal que codifica una coloración válida del grafo. Entonces $\sigma \in T^{n-1}$.*

Amortiguación de errores (I)

Operaciones moleculares del modelo:

- Son exactas.
- Están inspiradas en otras que son susceptibles de ser ejecutadas en el laboratorio (existencia de errores).

Operaciones conflictivas:

- `extraer` en el modelo restringido.
- `quitar` en el modelo débil.

A continuación se analiza un procedimiento que permite atenuar el efecto negativo de los errores cometidos en la operación `extraer`.

Amortiguación de errores (II)

Al implementar la operación **extraer**(T, s), a través de la manipulación de moléculas de ADN, puede suceder que:

- alguna molécula del tubo T contenga al símbolo s y, en cambio, pertenezca al tubo $-(T, s)$; o bien que
- alguna molécula del tubo T no contenga al símbolo s y, en cambio, pertenezca al tubo $+(T, s)$.

Los errores del primer tipo son *subsanables*.

Los errores del segundo tipo son *muy graves*.

A continuación vamos a ver cómo se pueden amortiguar los errores de este último tipo.

Amortiguación de errores (III)

La idea es la siguiente:

- A partir del tubo T, el símbolo s y un número natural n se procede como sigue:
  - Se ejecuta la operación `extraer` (T, s) que devuelve dos tubos: $+(T, s) = T_1$ y $-(T, s) = T'_1$.
  - Se ejecuta la operación `extraer` (T'_1, s) que devuelve dos tubos: $+(T'_1, s) = T_2$ y $-(T'_1, s) = T'_2$.
  - El proceso se reitera n veces.
  - Finalmente se devuelve la unión de los tubos T_1, $T_2, \ldots, T_n$.

Amortiguación de errores (IV)

Las ideas anteriores pueden ser descritas a través del siguiente programa molecular que denominaremos `extraer` $^{(n)}$:

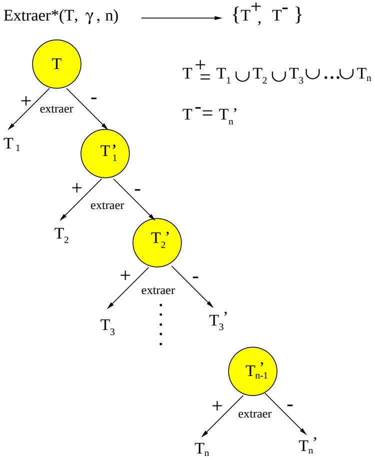
$$
\begin{array}{l}
\texttt{Entrada:}\ \ (T, s) \\
\qquad T_0 \leftarrow \emptyset;\quad T'_0 \leftarrow T \\
\qquad \texttt{Para}\ i \leftarrow 1\ \texttt{hasta}\ n\ \texttt{hacer} \\
\qquad\qquad T_i \leftarrow +(T'_{i-1}, s);\ \ T'_i \leftarrow -(T'_{i-1}, s) \\
\qquad\qquad T^+ \leftarrow T_{i-1} \cup T_i \\
\qquad T^- \leftarrow T'_n
\end{array}
$$

La ejecución de este programa devuelve dos tubos, T^+ y T^-:

- La probabilidad de que una molécula de T que contiene a s esté en T^+ es $1 - (p_+)^n$.
  (p_+ es la probabilidad de que tras ejecutar la operación **extraer**(T, s), una molécula que debería estar en $+(T, s)$ no se encuentre en dicho tubo).

Por tanto, es posible amortiguar el error antes citado a unos niveles prefijados con tal de elegir n convenientemente.

Amortiguación de errores (V)



El problema de las familias disjuntas

Sea $A = \{1, \ldots, p\}$ un conjunto finito. Sea $\mathcal{F} = \{B_1, \ldots, B_q\}$ una familia finita de subconjuntos de A. Determinar todas las subfamilias de $\mathcal{F}$ cuyos elementos son disjuntos dos a dos.

El problema de las familias disjuntos es presuntamente intratable.

Notación: Para cada j $(1 \leq j \leq q)$, notaremos $r_j = |B_j|$ y $B_j = \{x_j^1, \ldots, x_j^{r_j}\}$

Una solución del problema de las familias disjuntas en el modelo sticker (I)

Idea de un programa molecular: a partir de una $(p+q,q)$–librería, T_0, que codifica todas las posibles subfamilias de $\mathcal{F}$, se procede así:

- Se clasifican las moléculas de T_0 entre las que contienen al 1 (T^+), y las que no (T^-). De entre las que contienen al 1:
  - ⋆ Si para algún $x_1^j \in B_1$ tiene activada la posición $q + x_1^j$, se **desecha**.
  - ⋆ Si no, se activan las posiciones $q + x_1^j$ (T^*).

  Se considera $T_0 = T^* \cup T^-$.
- Se reitera el proceso para 2 (respecto de B_2), ..., q (respecto de B_q).

Una solución del problema de las familias disjuntas en el modelo sticker (II)

Diseño de un programa molecular:

```
Entrada:  T_0 (una (p+q, q)-librería)
      para i ← 1 hasta q hacer
            T^+ ← +(T_0, i);  T^- ← −(T_0, i)
            T_0^* ← T^+
            para j ← 1 hasta r_i hacer
                  T_basura ← +(T_{j−1}^*,  q + x_i^j)
                  T ← −(T_{j−1}^*,  q + x_i^j)
                  T_j^* ← activar(T,  q + x_i^j)
            T_0 ← mezcla(T_{r_i}^*, T^−)
```

Verificación formal de dicho programa molecular (I)

Para establecer la verificación, se etiquetan los tubos que se obtienen a lo largo de la ejecución, reescribiendo el programa molecular como sigue:

$\texttt{Entrada:}\quad T_0$ $\texttt{(una}$ $(p+q,q)\texttt{-librería)}$

$\quad\quad\texttt{para } i \leftarrow 1 \texttt{ hasta } q \texttt{ hacer}$

$\quad\quad\quad\quad T_i^{+} + \leftarrow +(T_{i-1}, i)\texttt{;}\ \ T_i^{-} \leftarrow -(T_{i-1}, i)$

$\quad\quad\quad\quad T_{i,0}^{*} \leftarrow T_i^{+}$

$\quad\quad\quad\quad \texttt{para } j \leftarrow 1 \texttt{ hasta } r_i \texttt{ hacer}$

$\quad\quad\quad\quad\quad\quad T_{i,j}^{basura} \leftarrow +(T_{i,j-1}^{*},\ \ q+x_i^j)$

$\quad\quad\quad\quad\quad\quad T_{i,j}^{\bullet} \leftarrow -(T_{i,j-1}^{*},\ \ q+x_i^j)$

$\quad\quad\quad\quad\quad\quad T_{i,j}^{*} \leftarrow \texttt{activar}(T_{i,j}^{\bullet},\ \ q+x_i^j)$

$\quad\quad\quad\quad T_i \leftarrow \texttt{mezcla}(T_{i,r_i}^{*}, T_i^{-})$

Verificación formal de dicho programa molecular (II)

Notación:

- σ se identifica con un número binario de $q+p$ dígitos.
- Si $\sigma = \sigma(1)\ldots\sigma(q)\sigma(q+1)\ldots\sigma(q+p)$, entonces

$$\begin{cases} \sigma_q = \sigma(1)\ldots\sigma(q) \\ \sigma_p = \sigma(q+1)\ldots\sigma(q+p) \end{cases}$$

- Diremos que $k \in \sigma$ sii $\sigma(k) = 1$. Relación de inclusión entre moléculas.

Verificación formal de dicho programa molecular (III)

Interpretación de la ejecución del programa molecular:

- Evolución de una población de individuos a lo largo del tiempo.
- Al transcurrir una unidad de tiempo, un individuo puede:
  - ⋆ Morir.
  - ⋆ Mutar.
  - ⋆ Permanecer invariable.
- Evolución de una molécula en una unidad de tiempo: $\texttt{STEP}(\rho, i)$, para $\rho \in T_{i-1}$ y $1 \leq i \leq q$.
- Historia de $\sigma \in T_0$: $\widehat{\sigma} = (\sigma^0, \sigma^1, \ldots, \sigma^q)$.
  En donde $\sigma^0 = \sigma$ y $\sigma^i = \texttt{STEP}(\sigma^{i-1}, i)$.

Verificación formal de dicho programa molecular (IV)

Lema 1: $\forall i\ (1 \leq i \leq q \rightarrow \forall \rho \in T_{i-1}\ (\mathtt{STEP}(\rho, i) \downarrow \rightarrow \mathtt{STEP}(\rho, i) \in T_i))$.

Lema 2:
$\forall i \forall j (1 \leq i \leq q\ \wedge\ 1 \leq j \leq r_i \rightarrow \forall \tau \in T^*_{i,j} \exists \rho \in T^+_i (\tau_q = \rho_q \wedge \forall s((1 \leq s \leq j \rightarrow \rho(q + x^j_i) = 0 \wedge \tau(q + x^j_i) = 1) \wedge (j < s \leq q \rightarrow \rho(q + x^j_i) = \tau(q + x^j_i)))))$

Corolario 3:
$\forall i(1 \leq i \leq q \rightarrow \forall \tau \in T^*_{i,r_i} \exists \rho \in T_{i-1}(i \in \rho \wedge \rho \neq \tau \wedge \mathtt{STEP}(\rho, i) = \tau))$

Corolario 4: $\forall i \forall j\ (1 \leq i \leq q \wedge 1 \leq j \leq r_i \rightarrow T^*_{i,j} \cap T^-_i = \emptyset)$

Lema 5: $\forall i\ (1 \leq i \leq q \rightarrow \forall \tau \in T^-_i (\tau \in T_{i-1} \wedge \mathtt{STEP}(\tau, i) = \tau))$

Lema 6: $\forall i\ (1 \leq i \leq q \rightarrow \forall \rho, \tau \in T_{i-1}\ (\mathtt{STEP}(\rho, i) \downarrow = \mathtt{STEP}(\tau, i) \rightarrow \rho = \tau))$

Corolario 7: $\forall i\ (1 \leq i \leq q \rightarrow \forall \sigma, \rho \in T_0\ (\sigma^i \downarrow = \rho^i \rightarrow \sigma = \rho))$

Verificación formal de dicho programa molecular (V)

Lema 8: $\forall i\ (1 \leq i \leq q \rightarrow \forall \tau \in T_i \exists! \sigma \in T_0\ (\sigma^i = \tau))$

Lema 9: $\forall i\ (1 \leq i \leq q \rightarrow \forall \sigma \in T_0\ (\sigma^i \downarrow \rightarrow \sigma^i \in T_i))$

Lema 10: $\forall i \forall j\ (1 \leq i \leq q \wedge 1 \leq j \leq r_i \rightarrow \forall \tau \in T^*_{i,j}\ (i \in \tau))$

Proposición 11: $\forall i\ (0 \leq i < q \rightarrow \forall \sigma \in T_0(\sigma^{i+1} \downarrow \rightarrow \sigma^i_q = \sigma^{i+1}_q \wedge \sigma^i_p \subseteq \sigma^{i+1}_p))$

Proposición 12:
$\forall i \forall k(1 \leq i \leq q \wedge 1 \leq k < i \rightarrow \forall \sigma \in T_0(\sigma^i \downarrow \rightarrow \sigma^k_q = \sigma^i_q \wedge \sigma^k_p \subseteq \sigma^i_p)$

Verificación formal de dicho programa molecular (VI)

Invariante de la **corrección**

$$\boxed{\theta(i) \equiv \forall \tau \in T_i \forall k, k' \in \tau \; (1 \leq k < k' \leq i \rightarrow B_k \cap B_{k'} = \emptyset)}$$

Teorema 13: $\forall i \; (1 \leq i \leq q \rightarrow \theta(i))$.

Corolario 14: (Corrección)

$\forall \tau \in T_q \forall k, k' \in \tau \; (1 \leq k < k' \leq q \rightarrow B_k \cap B_{k'} = \emptyset)$

Invariante de la **completitud**

$$\boxed{\delta(i) \equiv \forall \sigma \in T_0 \; ((\forall k, k' \in \sigma \; (1 \leq k < k' \leq q \rightarrow B_k \cap B_{k'} = \emptyset)) \rightarrow \sigma^i \in T_i)}$$

Teorema 15: $\forall i \; (1 \leq i \leq q \rightarrow \delta(i))$.

Corolario 16: (Completitud)

$\forall \sigma \in T_0 \; ((\forall k, k' \in \tau \; (1 \leq k < k' \leq q \rightarrow B_k \cap B_{k'} = \emptyset)) \rightarrow \sigma^q \in T_q)$