Computación Bio–inspirada

Tema 5: Resolución eficiente de problemas NP-completos en modelos moleculares

David Orellana Martín
Mario de J. Pérez Jiménez

Grupo de Investigación en Computación Natural
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

dorellana@us.es (http://www.cs.us.es/∼dorellana/)
marper@us.es (http://www.cs.us.es/∼marper/)

Máster Universitario en Lógica, Computación e Inteligencia Artificial
Curso 2023-2024

Índice

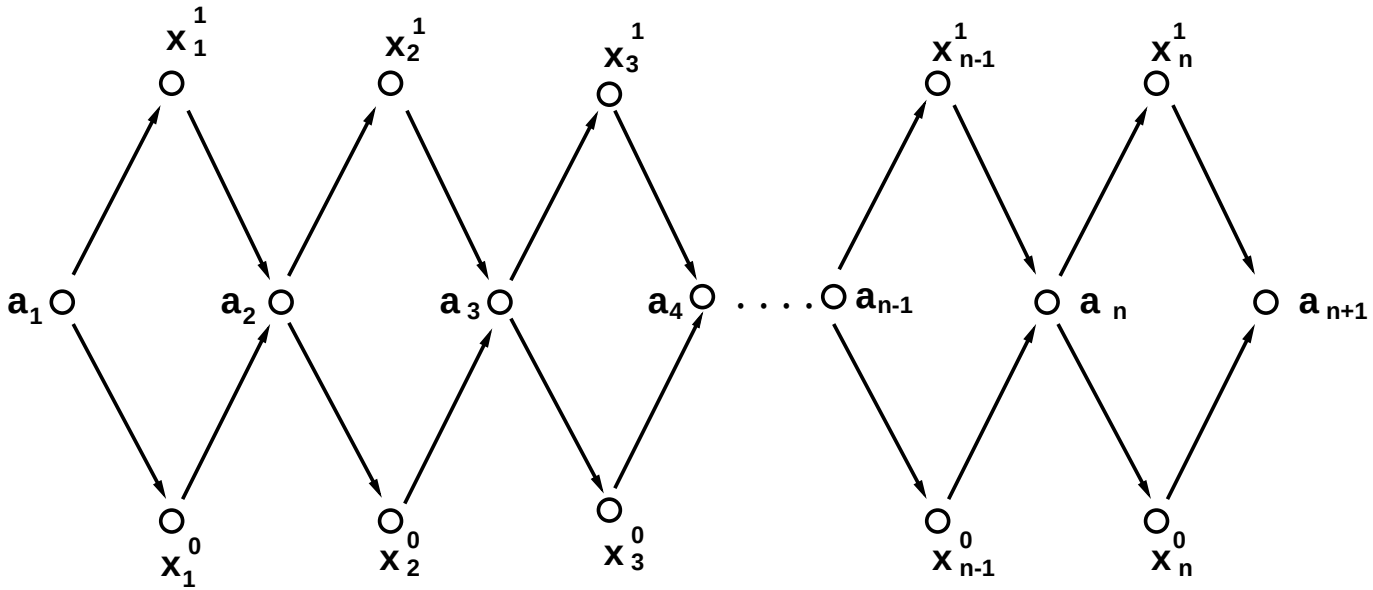
- El **problema SAT de la satisfactibilidad de la Lógica Proposicional**:
  - Una solución en el **modelo no restringido de Adleman**.
- El **problema** `3-COL`:
  - Una solución en el **modelo restringido de Adleman**.
  - Una solución en el **modelo débil de Amos**.
- El **problema de las familias disjuntas**:
  - Una solución en el **modelo sticker de Roweis**.

Una solución de SAT en el modelo no restringido

Sea $\varphi \equiv c_1 \wedge ... \wedge c_p$, con $c_i = l_{i,1} \vee ... \vee l_{i,r_i}$ una fórmula proposicional en FNC cuyo conjunto de variables es $\mathrm{Var}(\varphi) = \{x_1, ..., x_n\}$.

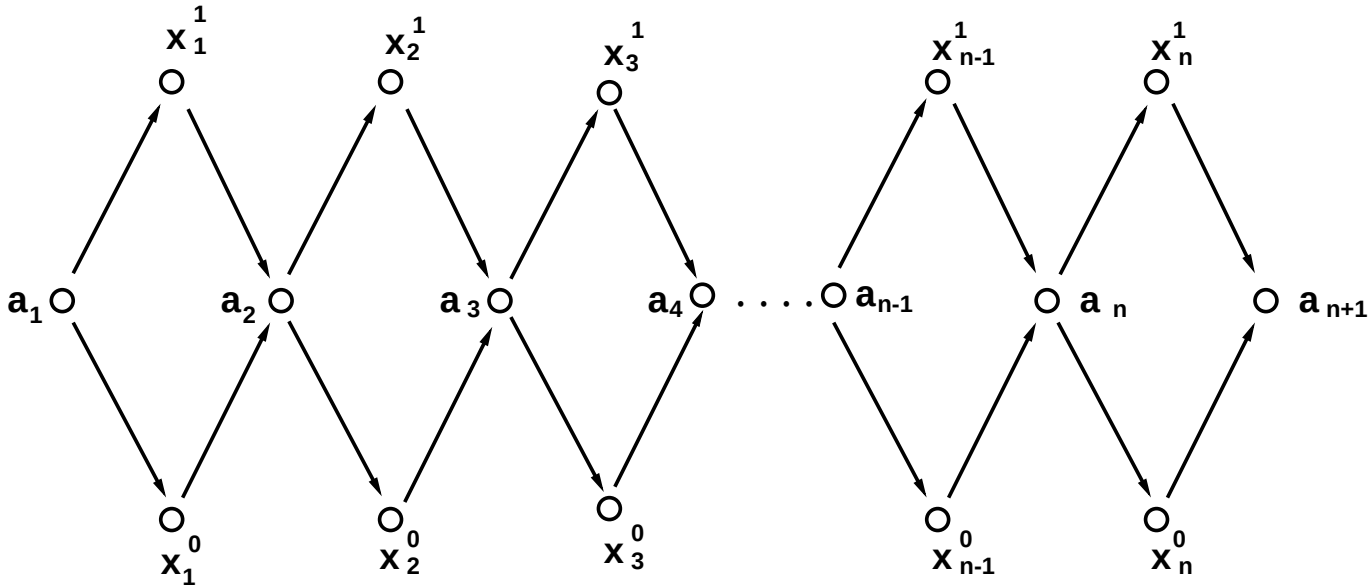
A la fórmula φ se le asocia el grafo dirigido $G_n = (V_n, E_n)$, definido como sigue:

- $V_n = \{a_i, x_i^j, a_{n+1} : \ 1 \leq i \leq n, \ 0 \leq j \leq 1\}$.
- $E_n = \{(a_i, x_i^j), (x_i^j, a_{i+1}) : \ 1 \leq i \leq n, \ 0 \leq j \leq 1\}$.



Una solución de SAT en el modelo no restringido

En el grafo G_n:



- ⋆ Existen 2^n caminos desde a_1 hasta a_{n+1}.
- ⋆ Cada camino $a_1 x_1^{i_1} a_2 x_2^{i_2} \ldots a_n x_n^{i_n} a_{n+1}$ desde a_1 hasta a_{n+1}, tiene asociado la valoración σ definida por: $\sigma(x_i) = j_i$ $(1 \leq i \leq n)$.

A efectos de una implementación biológica, con el grafo G_n se procedería como en el experimento de Adleman.

Un programa molecular en el modelo no restringido

Alfabeto: $\Sigma = \{a_i, x_i^{j_i}, a_{n+1} : \ 1 \leq i \leq n \ \wedge \forall i \ (1 \leq i \leq n \rightarrow (j_i = 0 \vee j_i = 1))\}$.

Tubo de entrada, $T_0 = \{a_1 x_1^{j_1} a_2 x_2^{j_2} \ldots a_n x_n^{j_n} a_{n+1} : \ \forall i \ (1 \leq i \leq n \rightarrow (j_i = 0 \vee j_i = 1))\}$

- Cadenas de Σ de longitud $2n + 1$.
- Cada "molécula" de T_0 representa/codifica una valoración relevante para la fórmula φ.

Para cada literal, $l_{i,j}$, que aparece en la fórmula:

- Si $l_{i,j} = x_m$, entonces notaremos $l_{i,j}^1 = x_m^1$, $l_{i,j}^0 = x_m^0$.
- Si $l_{i,j} = \overline{x}_m$, entonces notaremos $l_{i,j}^1 = x_m^0$, $l_{i,j}^0 = x_m^1$.

Es decir, si una molécula contiene $l_{i,j}^1$ (resp. $l_{i,j}^0$), entonces el literal $l_{i,j}$ tiene asignado el valor 1 (resp. 0) por la valoración que codifica dicha molécula.

Consideremos el siguiente programa molecular que resuelve el problema SAT:

```
Entrada:  T_0
    Para i ← 1 hasta p hacer
        T_1 ← T_0 ;   T_0 ← ∅
        Para j ← 1 hasta r_i hacer
            T' ← +(T_1, l^1_{i,j})
            T_1 ← −(T_1, l^1_{i,j})
            T_0 ← T_0 ∪ T'
    Detectar(T_0)
```

```
Entrada:  T_0
    Para i ← 1 hasta p hacer
        T_1 ← T_0 ;   T_0 ← ∅
        Para j ← 1 hasta r_i hacer
            T' ← +(T_1, l^1_{i,j})
            T_1 ← −(T_1, l^1_{i,j})
            T_0 ← T_0 ∪ T'
    Detectar(T_0)
```

Idea del programa molecular: a partir del tubo inicial, T_0, que representa/codifica todas las valoraciones relevantes para la fórmula de entrada, se procede como sigue:

- Se elabora un nuevo tubo, T_1, seleccionando de T_0 todas las valoraciones que hacen verdadera c_1:
  - ⋆ Se eligen las que hacen verdadero el literal $l_{1,1}$.
  - ⋆ De las que hacen falso $l_{1,1}$, se eligen las que hacen verdadero el literal $l_{1,2}$.
  - ⋆ De las que hacen falso $l_{1,1} \vee l_{1,2}$, se eligen las que hacen verdadero el literal $l_{1,3}$.
  - ⋆ Y así sucesivamente con todos los literales de c_1.
- Se elabora un nuevo tubo, T_2, seleccionando de T_1 todas las valoraciones que hacen verdadera c_2:
  - ⋆ Se eligen las que hacen verdadero el literal $l_{2,1}$.
  - ⋆ De las que hacen falso $l_{2,1}$, se eligen las que hacen verdadero el literal $l_{2,2}$.
  - ⋆ De las que hacen falso $l_{2,1} \vee l_{2,2}$, se eligen las que hacen verdadero el literal $l_{2,3}$.
  - ⋆ Y así sucesivamente con todos los literales de c_2.
- El proceso se reitera p veces hasta obtener el tubo de salida T_p, a partir de T_{p-1}, que contiene todas las valoraciones que hacen verdadera la fórmula $c_1 \wedge \cdots \wedge c_p$.

Una metodología para demostrar que un programa con un bucle principal satisface una propiedad

Búsquese una fórmula que satisfaga las condiciones siguientes:

- Está expresada en función de la variable del bucle.
- Es un **invariante** de dicho bucle.
  - Tras la ejecución de cada paso del bucle, la fórmula ha de ser verdadera (suele probarse por inducción acotada).
- De la veracidad de la fórmula tras la ejecución del bucle, se debe inferir que el programa verifica la propiedad requerida.

Importante: Para verificar formalmente un programa molecular (de un modelo basado en filtrados) hay que probar la **corrección** y **completitud** del mismo.

Verificación formal: Reetiquetado

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

```
Entrada:  T_0
    Para i ← 1 hasta p hacer
        T_1 ← T_0 ;   T_0 ← ∅
        Para j ← 1 hasta r_i hacer
            T' ← +(T_1, I^1_{i,j})
            T_1 ← −(T_1, I^1_{i,j})
            T_0 ← T_0 ∪ T'
    Detectar(T_0)
```

↓

```
Entrada:  T_0
    Para i ← 1 hasta p hacer
        T_{i,0} ← ∅ ;   T''_{i,0} ← T_{i−1}
        Para j ← 1 hasta r_i hacer
            T'_{i,j} ← +(T''_{i,j−1}, I^1_{i,j})
            T''_{i,j} ← −(T''_{i,j−1}, I^1_{i,j})
            T_{i,j} ← T_{i,j−1} ∪ T'_{i,j}
        T_i ← T_{i,r_i}
    Detectar(T_p)
```

Verificación formal: Corrección

Corrección del programa: se ha de probar que

- **Toda molécula del tubo de salida representa/codifica una valoración que hace verdadera la fórmula φ.**

O lo que es lo mismo:

- **Si el programa responde SÍ, entonces la fórmula φ debe ser satisfactible.**

Entonces:

- Para cada i $(1 \leq i \leq p)$, se notará $\varphi_i \equiv c_1 \wedge \ldots \wedge c_i$ (φ_0 es una tautología).
- Para cada i, j tales que $1 \leq i \leq p,\ 1 \leq j \leq r_i$, se notará $L_{i,j} \equiv l_{i,1} \vee \ldots \vee l_{i,j}$.
- Para cada i, j tales que $1 \leq i \leq p,\ 1 \leq j \leq r_i$, se define

$$\psi(i,j) \equiv \forall \sigma \in T_{i,j}\ (\sigma(\varphi_{i-1}) = 1 \wedge \sigma(L_{i,j}) = 1) \wedge \forall \sigma \in T''_{i,j}\ (\sigma(\varphi_{i-1}) = 1 \wedge \sigma(L_{i,j}) = 0)$$

- Para cada i $(1 \leq i \leq p)$, se define $\theta(i) \equiv \forall j\ (1 \leq j \leq r_i \rightarrow \psi(i,j))$.

$\theta(i) \equiv$ **toda molécula de T_i representa/codifica una valoración que hace verdadera φ_i**

Verificación formal: Corrección

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

Demostración por inducción sobre i.

- Caso base: $i = 1$

  Veamos que la fórmula $\theta(1) \equiv \forall j\ (1 \leq j \leq r_1 \rightarrow \psi(1, j))$ es verdadera.
  A su vez, demostraremos este resultado por inducción sobre j

  * Caso base: $j = 1$ Veamos que es verdadera la fórmula $\psi(1,1) \equiv$ $\forall \sigma \in T_{1,1}\ (\sigma(\varphi_0) = 1 \wedge \sigma(L_{1,1}) = 1) \wedge \forall \sigma \in T''_{1,1}\ (\sigma(\varphi_0) = 1 \wedge \sigma(L_{1,1}) = 0)$. En efecto:
    * Por una parte, si $\sigma \in T_{1,1} = T_{1,0} \cup T'_{1,1} = T'_{1,1} = +(T''_{1,0}, I^1_{1,1}) = +(T_0, I^1_{1,1})$ entonces $\sigma \in T_0$ y $\sigma(I_{1,1}) = 1$. Luego, $\sigma(\varphi_0) = 1 \wedge \sigma(L_{1,1}) = 1$.
    * Por otra, si $\sigma \in T''_{1,1} = -(T''_{1,0}, I^1_{1,1}) = -(T_0, I^1_{1,1})$ entonces $\sigma \in T_0$ y $\sigma(I_{1,1}) = 0$. Luego, $\sigma(\varphi_0) = 1$ y $\sigma(L_{1,1}) = \sigma(I_{1,1}) = 0$.

  * Paso inductivo: Sea $j \geq 1$ tal que $j < r_1$ y que $\psi(1, j)$ es verdadera. Entonces hemos de ver que también es verdadera la fórmula $\psi(1, j+1)$ siguiente:

    $\forall \sigma \in T_{1,j+1}\ (\sigma(\varphi_0) = 1 \wedge \sigma(L_{1,j+1}) = 1) \wedge \forall \sigma \in T''_{1,j+1}\ (\sigma(\varphi_0) = 1 \wedge \sigma(L_{1,j+1}) = 0)$.
    * Por una parte, si $\sigma \in T_{1,j+1} = T_{1,j} \cup T'_{1,j+1} = T_{1,j} \cup +(T''_{1,j}, I^1_{1,j+1})$ entonces: o bien, $\sigma \in T_{1,j}$, en cuyo caso por H.I. $\sigma(\varphi_0 \wedge L_{1,j}) = 1$ (luego, $\sigma(\varphi_0) = 1$ y $\sigma(L_{1,j+1}) = 1$); o bien $\sigma \in +(T''_{1,j}, I^1_{1,j+1})$, en cuyo caso, por H.I. $\sigma(\varphi_0) = 1$ y $\sigma(L_{1,j}) = 0$ (luego, $\sigma(\varphi_0) = 1 \wedge \sigma(L_{1,j+1}) = 1$).
    * Por otra, si $\sigma \in T''_{1,j+1} = -(T''_{1,j}, I^1_{1,j+1})$ entonces $\sigma \in T''_{1,j}$ y $\sigma(I_{1,j+1}) = 0$. Como por H.I. se tiene que $\sigma(\varphi_0) = 1 \wedge \sigma(L_{1,j}) = 0$, resulta que $\sigma(\varphi_0) = 1 \wedge \sigma(L_{1,j+1}) = 0$.

Verificación formal: Corrección

- **Paso inductivo** Sea $i \geq 1$ tal que $i < p$ y $\theta(i)$ es verdadera Entonces hemos de ver que la fórmula $\theta(i+1) \equiv \forall j\ (1 \leq j \leq r_{i+1} \rightarrow \psi(i+1, j))$ tambien es verdadera.

  A su vez, demostraremos este resultado por inducción sobre j

  - **Caso base:** $j = 1$ Veamos que es verdadera la fórmula $\psi(i+1, 1)$ siguiente
    $\forall \sigma \in T_{i+1,1}\ (\sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,1}) = 1) \wedge \forall \sigma \in T''_{i+1,1}\ (\sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,1}) = 0)$.

    En efecto:

    - Por una parte, si

      $\sigma \in T_{i+1,1} = T_{i+1,0} \cup T'_{i+1,1} = T'_{i+1,1} = +(T''_{i+1,0}, l^1_{i+1,1}) = +(T_i, l^1_{i+1,1})$ entonces $\sigma \in T_i = T_{i,r_i}$ y $\sigma(l_{i+1,1}) = 1$. Por tanto, $\sigma(\varphi_{i-1}) = 1 \wedge \sigma(L_{i,r_i}) = 1$ (por H.I.) y $\sigma(l_{i+1,1}) = 1$; es decir, $\sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,1}) = 1$.
    - Por otra, si $\sigma \in T''_{i+1,1} = -(T''_{i+1,0}, l^1_{i+1,1}) = -(T_i, l^1_{i+1,1})$ entonces $\sigma \in T_i$ y $\sigma(l_{i+1,1}) = 0$. Luego, $\sigma(\varphi_i) = 1$ (por H.I.) y $\sigma(L_{i+1,1}) = 0$.

  - **Paso inductivo:** Sea $j \geq 1$ tal que $j < r_1$ y que $\psi(i+1, j)$ es verdadera. Hemos de ver que la fórmula $\psi(i+1, j+1) \equiv \forall \sigma \in T_{i+1,j+1}\ (\sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,j+1}) = 1) \wedge$
    $\forall \sigma \in T''_{i+1,j+1}\ (\sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,j+1}) = 0)$ también es verdadera.
    - Por una parte, si $\sigma \in T_{i+1,j+1} = T_{i+1,j} \cup T'_{i+1,j+1}$ entonces: o bien, $\sigma \in T_{i+1,j}$, en cuyo caso por H.I. $\sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,j}) = 1$ (luego, $\sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,j+1}) = 1$); o bien $\sigma \in T'_{i+1,j+1} = +(T''_{i+1,j}, l^1_{i+1,j+1})$, en cuyo caso, $\sigma \in T''_{i+1,j}$ y $\sigma(l_{i+1,j+1}) = 1$, luego $\sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,j+1}) = 1$.
    - Por otra, si $\sigma \in T''_{i+1,j+1} = -(T''_{i+1,j}, l^1_{i+1,j+1})$ entonces $\sigma \in T''_{i+1,j}$ y $\sigma(l_{i+1,j+1}) = 0$. Por H.I. se tiene que $\sigma \in T''_{i+1,j} \Rightarrow \sigma(\varphi_i) = 1 \wedge \sigma(L_{i+1,j}) = 0$. En consecuencia, se deduce que $\sigma(\varphi_i) = 1$ y $\sigma(L_{i+1,j+1}) = 0$.

Verificación formal: Corrección

Corolario: (**Corrección** del programa)
Toda molécula del tubo de salida representa/codifica una valoración que hace verdadera φ.

Demostración: Del teorema anterior resulta que la fórmula $\theta(p)$ es verdadera y, en consecuencia, toda molécula del tubo de salida T_p representa una valoración de verdad que asigna el valor 1 a φ_p; es decir, a la fórmula φ.

■

Verificación formal: Completitud

Completitud del programa: se ha de probar que

- **Toda molécula del tubo de entrada que codifique una valoración que hace verdadera φ, deberá aparecer, también, en el tubo de salida.**

O lo que es lo mismo, se ha de probar que:

- **Si el programa responde NO, entonces la fórmula φ no es satisfactible**.

Se considera la siguiente fórmula:

$$\delta(i) \equiv \forall \sigma \in T_0 \ (\sigma(\varphi) = 1 \rightarrow \sigma \in T_i)$$

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

Es decir, $\delta(i) \equiv$ toda molécula del tubo inicial que codifica una valoración que hace verdadera φ, está en el tubo T_i.

```
Entrada:  T_0
    Para i ← 1 hasta p hacer
        T_{i,0} ← ∅;   T''_{i,0} ← T_{i-1}
        Para j ← 1 hasta r_i hacer
            T'_{i,j} ← +(T''_{i,j-1}, l^1_{i,j})
            T''_{i,j} ← −(T''_{i,j-1}, l^1_{i,j})
            T_{i,j} ← T_{i,j-1} ∪ T'_{i,j}
        T_i ← T_{i,r_i}
    Detectar(T_p)
```

Verificación formal: Completitud

Teorema: La fórmula $\delta(i)$ es un invariante del bucle principal del programa; es decir, $\forall i\ (1 \leq i \leq p \rightarrow \delta(i))$.

Demostración por inducción sobre i.

- Caso base: $i = 1$

  Veamos que la fórmula $\delta(1) \equiv \forall \sigma \in T_0\ (\sigma(\varphi) = 1 \rightarrow \sigma \in T_1)$ es verdadera.
  Comencemos observando que si $\sigma \in T_0$ es tal que $\sigma(\varphi) = 1$, entonces $\sigma(c_1) = \ldots = \sigma(c_p) = 1$.
  Para cada t $(1 \leq t \leq p)$, sea k_t =mín $\{j \mid 1 \leq j \leq r_t \wedge \sigma(l_{t,j}) = 1\}$; es decir, k_t es el menor índice de un literal de c_t verdadero por σ. Luego, $\forall j(1 \leq j < k_t \Rightarrow \sigma(l_{t,j}) = 0)$.
  Para ver que $\delta(1)$ es verdadera, sea $\sigma \in T_0$ tal que $\sigma(\varphi) = 1$.

  * Caso 1: $k_1 = 1$. En este caso, $\sigma \in T_0 \subseteq T''_{1,0} \wedge \sigma(l_{1,1}) = 1$. Luego, $\sigma \in +(T''_{1,0}, l_{1,1}) = T'_{1,1} \subseteq T_{1,1} \subseteq T_1$.

  ```
  Entrada: T_0
      Para i ← 1 hasta p hacer
          T_{i,0} ← ∅;   T''_{i,0} ← T_{i−1}
          Para j ← 1 hasta r_i hacer
              T'_{i,j} ← +(T''_{i,j−1}, l^1_{i,j})
              T''_{i,j} ← −(T''_{i,j−1}, l^1_{i,j})
              T_{i,j} ← T_{i,j−1} ∪ T'_{i,j}
          T_i ← T_{i,r_i}
      Detectar(T_p)
  ```

  * Caso 2: $k_1 > 1$. En este caso, $\sigma \in T_0 \subseteq T''_{1,0} \wedge \sigma(l_{1,k_1-1}) = 0$. Luego, $\sigma \in -(T''_{1,0}, l_{1,k_1-1}) = T''_{1,k_1-1}$. Puesto que $\sigma(l_{1,k_1}) = 1$, se tiene que $\sigma \in +(T''_{1,k_1-1}, l_{1,k_1}) = T'_{1,k_1} \subseteq T_1$.

Verificación formal: Completitud

- **Paso inductivo:** Sea $i, 1 \leq i < p$, tal que la fórmula $\delta(i)$ es verdadera. Veamos que la fórmula $\delta(i+1) \equiv \forall \sigma \in T_0\ (\sigma(\varphi) = 1 \rightarrow \sigma \in T_{i+1})$ también es verdadera.
  Para ello, sea $\sigma \in T_0$. Por H.I. la fórmula $\delta(i)$ es verdadera y, por tanto, $\sigma \in T_i = T''_{i+1,0}$

  * **Caso 1:** $k_{i+1} = 1$. En este caso, $\sigma \in T''_{i+1,0} \wedge \sigma(l_{i+1,1}) = 1$. Luego, $\sigma \in +(T''_{i+1,0}, l_{i+1,1}) = T'_{i+1,1} \subseteq T_{i+1,1} \subseteq T_{i+1}$.

```
Entrada:  T_0
    Para i ← 1 hasta p hacer
        T_{i,0} ← ∅;   T''_{i,0} ← T_{i-1}
        Para j ← 1 hasta r_i hacer
            T'_{i,j} ← +(T''_{i,j-1}, l^1_{i,j})
            T''_{i,j} ← −(T''_{i,j-1}, l^1_{i,j})
            T_{i,j} ← T_{i,j-1} ∪ T'_{i,j}
        T_i ← T_{i,r_i}
    Detectar(T_p)
```

  * **Caso 2:** $k_{i+1} > 1$. En este caso, $\sigma \in T''_{i+1,0} \wedge \forall j\,(1 \leq j < k_{i+1} \Rightarrow \sigma(l_{i+1,j}) = 0)$. De donde se tiene que $\forall j\,(1 \leq j < k_{i+1} \Rightarrow \sigma \in T''_{i+1,j})$. En particular $\sigma \in T''_{i+1,k_{i+1}-1}$. Puesto que $\sigma(l_{i+1,k_{i+1}}) = 1$ se deduce que $\sigma \in +(T''_{i+1,k_{i+1}-1}, l_{i+1,k_{i+1}}) = T'_{i+1,k_{i+1}} \subseteq T_{i+1,k_{i+1}} \subseteq T_{i+1}$.

Verificación formal: Completitud

Corolario: (**Completitud** del programa)
Toda molécula del tubo de entrada que representa/codifica una valoración que hace verdadera φ, pertenece al tubo de salida.

Demostración: Del teorema anterior se deduce que es verdadera la fórmula $\delta(p) \equiv \forall \sigma \in T_0 \ (\sigma(\varphi) = 1 \rightarrow \sigma \in T_p)$

Por tanto, si $\sigma \in T_0$ es tal que $\sigma(\varphi) = 1$ entonces $\sigma \in T_p$; es decir, σ pertenece al tubo de salida T_p.

■

Una solución de 3-COL en el modelo restringido

Sea $G = (V, E)$ un grafo no dirigido, con $V = \{1, ..., n\}$. Notaremos:

* $p_1, \ldots, p_n$: códigos moleculares de los nodos.
* c_1, c_2, c_3: códigos moleculares de los colores.
* $\{e_1, e_2, \ldots, e_p\}$: conjunto ordenado de aristas de G.
* $e_i = \{e_i^1, e_i^2\}$, con $e_i^1 < e_i^2$.

Se considera el alfabeto:

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

Cada símbolo $(p_1, x_1, p_2, x_2, ..., p_n, x_n)$ del alfabeto Σ se identifica con una molécula $p_1x_1p_2x_2...p_nx_n$ de ADN que representa una **coloración del grafo** codificada por:

* **El color del nodo i es x_i (para cada i, $1 \leq i \leq n$).**

Si $\sigma = p_1x_1p_2x_2...p_nx_n$, notaremos $(\sigma)_i = x_i$. Asímismo, notaremos $p_i(c_j)$ para indicar que el nodo p_i está coloreado con el color c_j.

El tubo de entrada T_0 del programa va a ser el propio alfabeto Σ (que codifica todas las posibles coloraciones del grafo).

Un programa molecular en el modelo restringido

Consideremos el siguiente programa molecular que resuelve el problema 3-COL:

```
Entrada:  T_0
    Para i ← 1 hasta p hacer
        T_1 ← +(T_0, e_i^1(c_1));  T_1^* ← −(T_0, 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_0 ← T_1' ∪ T_2'
        T_0 ← T_0 ∪ T_3'
    Detectar(T_0)
```

Idea del programa molecular: a partir del tubo inicial T_0 que codifica todas las coloraciones relevantes del grafo de entrada, 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 c_j a e_1^1 y en T_j' las coloraciones de T_j que dan a e_1^1 un color distinto de c_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).

Verificación formal (I)

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)

$T^0 \leftarrow T$

Para $i \leftarrow 1$ hasta p hacer

$T_{i,1} \leftarrow +(T^{i-1}, e_i^1(c_1))$; $T^*_{i,1} \leftarrow -(T^{i-1}, e_i^1(c_1))$

$T_{i,2} \leftarrow +(T^*_{i,1}, e_i^1(c_2))$; $T_{i,3} \leftarrow -(T^*_{i,1}, e_i^1(c_2))$

Para $j \leftarrow 1$ hasta 3 hacer

$T'_{i,j} \leftarrow -(T_{i,j}, e_i^2(c_j))$

$\overline{T}^i \leftarrow T'_{i,1} \cup T'_{i,2}$

$T^i \leftarrow \overline{T}^i \cup T'_{i,3}$

Detectar(T^p)

Verificación formal (II)

Corrección del programa: se ha de probar que

- **Toda molécula del tubo de salida codifica una coloración válida del grafo con tres colores.**

O lo que es lo mismo, se ha de probar que:

- **Si el programa responde SÍ, entonces el grafo G es coloreable con tres colores**.

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 $1 \leqslant i \leqslant p$.

Obsérvese que $\theta(i)$ es verdadera sii 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 (III)

Lema 1: Para cada i, j $(1 \leq i \leq p \ \wedge \ 1 \leq j \leq 3)$ se verifica:

- $T_{i,j} \subseteq T^{i-1}$ y $T'_{i,j} \subseteq T^{i-1}$.
- Para cada molécula $\sigma \in T_{i,j}$ se tiene que $(\sigma)_{e_i^1} = c_j$.
- Para cada molécula $\sigma \in T'_{i,j}$ se tiene que $(\sigma)_{e_i^1} = c_j$ y $(\sigma)_{e_i^2} \neq c_j$.

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

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

- ⋆ Prueba por inducción sobre i.

Corolario: (**Corrección** del programa)
Toda molécula del tubo de salida T^p codifica una coloración válida del grafo.

Verificación formal (IV)

Completitud del programa: se ha de probar que

- **Toda molécula del tubo inicial que codifique una coloración válida del grafo con tres colores, aparecerá en el tubo de salida.**

O lo que es lo mismo, se ha de probar que:

- **Si el programa responde NO, entonces el grafo G no es coloreable con tres colores**.

Para probar la completitud del programa se considera la fórmula:

$$\delta(i) \equiv \forall \sigma \in T^0([\forall k \leq p\ ((\sigma)_{e_k^1} \neq (\sigma)_{e_k^2})] \rightarrow \sigma \in T^i)$$

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

$\delta(i)$ es verdadera sii toda molécula del tubo inicial que codifica una coloración válida del grafo con tres colores, pertenece al tubo relevante T^i, que se obtiene a lo largo de la ejecución del programa.

Teorema: La fórmula $\delta(i)$ es un **invariante** del bucle principal del programa; es decir,
$\forall \sigma\ (\ \underbrace{\sigma \in T^0 \wedge [\forall k \leq p\ ((\sigma)_{e_k^1} \neq (\sigma)_{e_k^2})]}\ \longrightarrow\ \underbrace{\forall i\ (1 \leq i \leq p \rightarrow \sigma \in T^i)}\ )$.

- ⋆ Prueba por inducción sobre i.

Corolario: (**Completitud** del programa)
Toda molécula del tubo inicial que codifica una coloración válida del grafo, pertenece al tubo T^p de salida del programa

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 $i \leftarrow 1$ hasta $n-1$ hacer
copiar$(T, \{T_1, T_2, T_3\})$
Para $j \leftarrow 1$ hasta 3 hacer
quitar$(T_j, \{x_i \neq j, p_k(c_j) : \ k > i \wedge \{i,k\} \in E\})$
union$(\{T_1, T_2, T_3\}, T)$
Seleccion(T)

Verificación formal (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($T_j^{i-1}, \{x_i \neq j, p_k(c_j) : \quad k > i \wedge \{i,k\} \in E\}$)
union($\{\overline{T}_1^i, \overline{T}_2^i, \overline{T}_3^i\}, T^i$)
Seleccion(T^{n-1})

Verificación formal (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 $\sigma \in \overline{T}^i_j$ se verifica que $(\sigma)_i = j \wedge \forall k \ (i < k \wedge \{i,k\} \in E \rightarrow (\sigma)_k \neq j)$.

Verificación formal (III)

Corrección del programa: se ha de probar que

- **Toda molécula del tubo de salida codifica una coloración válida del grafo con tres colores.**

O lo que es lo mismo, se ha de probar que:

- **Si el programa responde SÍ, entonces el grafo G es coloreable con tres colores**.

Para probar la corrección del programa molecular, para cada i $(1 \leq i \leq n-1)$ se considera la fórmula:

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

$\theta(i)$ es verdadera **sii** 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: La fórmula $\theta(i)$ 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 T^{n-1} codifica una coloración válida del grafo.

Verificación formal (IV)

Completitud del programa: se ha de probar que

- **Toda molécula del tubo inicial que codifique una coloración válida del grafo con tres colores, aparecerá en el tubo de salida.**

O lo que es lo mismo, se ha de probar que:

- **Si el programa responde NO, entonces el grafo G no es coloreable con tres colores**.

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 8: (**Completitud** del programa)
Toda molécula del tubo inicial que codifica una coloración válida del grafo, aparece en el tubo de salida T^{n-1}.

Amortiguación de errores (I)

Operaciones moleculares del **modelo formal**:

- Son **exactas**.
- Están inspiradas en operaciones susceptibles de ser ejecutadas en el laboratorio (existencia de **errores**).

Operaciones conflictivas:

- **extraer** en el modelo restringido o no 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, γ) (o **extraer**(T, s)), a través de la manipulación de moléculas de ADN, puede suceder:

- o bien que alguna molécula del tubo T contenga la cadena γ y, en cambio, pertenezca al tubo $-(T, \gamma)$;
- o bien que alguna molécula del tubo T **no** contenga a la cadena γ y, en cambio, pertenezca al tubo $+(T, \gamma)$.

Los errores del segundo tipo son **subsanables**.

Los errores del primer tipo son **muy graves**.

Veamos cómo se pueden "amortiguar" los errores de este último tipo.

Amortiguación de errores (III)

La idea es la siguiente:

- A partir del tubo T, la cadena γ y un número natural n se procede como sigue:
  - Se ejecuta la operación `extraer` (T, γ) que devuelve dos tubos: $+(T, \gamma) = T_1$ y $-(T, \gamma) = T'_1$.
  - Se ejecuta la operación `extraer` (T'_1, γ) que devuelve dos tubos: $+(T'_1, \gamma) = T_2$ y $-(T'_1, \gamma) = T'_2$.
  - 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 podríamos denominar **extraer** $^{(n)}$):

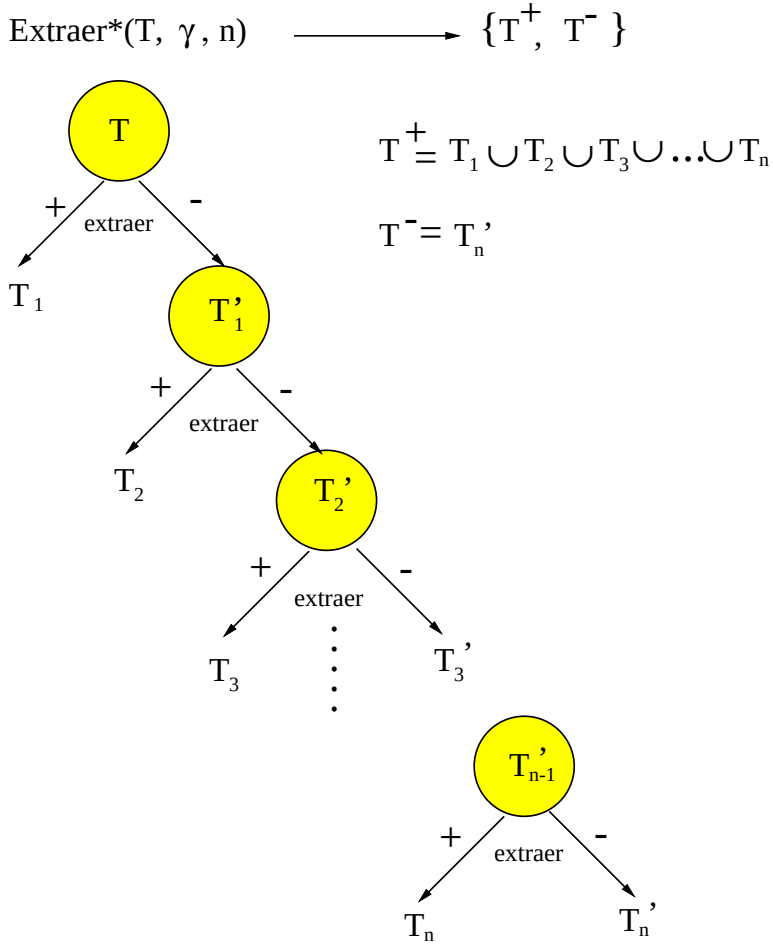
```
Entrada:  (T, γ)
          T_0 ← ∅;   T'_0 ← T
          Para i ← 1 hasta n hacer
                T_i ← +(T'_{i-1}, γ);  T'_i ← −(T'_{i-1}, γ)
                T^+ ← T_{i-1} ∪ T_i
          T^− ← T'_n
```

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

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

Conclusión: Es posible amortiguar el error del primer tipo a unos niveles prefijados, con tal de elegir n convenientemente.

Amortiguación de errores (V)



El problema de las familias disjuntas

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

Este problema 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 $(p+q, q)$–biblioteca está formada por complejos que codifican **todos** los números binarios con $(p+q)$ dígitos tales que: los q primeros se consideran de todas las formas posibles y los $(p+q) - q = p$ últimos son nulos (en total 2^q números binarios).

Cada complejo de esa biblioteca **representará** una **subfamilia** $\mathcal{F}'$ de $\mathcal{F}$:

- Si en la posición j $(1 \leq j \leq q)$ aparece un 1 significa que el conjunto B_j pertenece a la subfamilia representada por ese complejo. Si aparece un 0, no pertenece a esa subfamilia.

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

Idea de un programa molecular: a partir de una $(p+q, q)$–biblioteca, 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 (tubo T^+), y las que no (tubo 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 $q + x_1^j$ (tubo T^*).

  Se considera $T_0 = T^* \cup T^-$.

- Se reitera el proceso para 2 (respecto de B_2), ..., para 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 $i \leftarrow 1$ hasta q hacer
$T^+ \leftarrow +(T_0, i)$; $T^- \leftarrow -(T_0, i)$
$T_0^* \leftarrow T^+$
para $j \leftarrow 1$ hasta r_i hacer
$T_{basura} \leftarrow +(T_{j-1}^*, \ q + x_i^j)$
$T \leftarrow -(T_{j-1}^*, \ q + x_i^j)$
$T_j^* \leftarrow \mathtt{activar}(T, \ q + x_i^j)$
$T_0 \leftarrow \mathtt{mezcla}(T_{r_i}^*, T^-)$

Verificación formal (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 (una $(p+q,q)$-librería)

para $i \leftarrow 1$ hasta q hacer

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

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

para $j \leftarrow 1$ hasta r_i hacer

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

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

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

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

Verificación formal (II)

Notación:

- σ se identifica con un número binario de $p+q$ 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 (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 (IV)

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

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

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

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

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

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

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

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 (V)

Fórmula invariante de la **corrección**

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

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

Corolario 14: (Corrección)
$\forall \tau \in T_q \forall k, k' \in \tau \ (1 \leq k < k' \leq q \rightarrow B_k \cap B_{k'} = \emptyset)$

Fórmula invariante de la **completitud**

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

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

Corolario 16: (Completitud)
$\forall \sigma \in T_0 \ ((\forall k, k' \in \tau \ (1 \leq k < k' \leq q \rightarrow B_k \cap B_{k'} = \emptyset)) \rightarrow \sigma^q \in T_q)$