

1

Computación molecular sin memoria basada en ADN

MARIO DE JESÚS PÉREZ JIMÉNEZ

1.1 Introducción

A principio de la década de los cincuenta comienza a ponerse de manifiesto la analogía existente entre algunos procedimientos matemáticos y ciertos procesos biológicos: por una parte, para calcular el valor de una función computable en un elemento, x , de su dominio, hay que aplicar una serie de operaciones elementales del modelo (en el que la función es computable) al dato de entrada x ; por otra, todo organismo vivo se puede considerar, en esencia, como el resultado de un proceso que consiste en aplicar una serie de operaciones bioquímicas, a una cadena de ácido desoxirribonucleico (ADN).

L.M. Adleman materializó esta similitud en noviembre de 1994 ([1]) mostrando que era posible usar procesos biológicos para atacar la resolubilidad de ciertos problemas matemáticos especialmente *difíciles*: mediante

Mario de Jesús Pérez Jiménez. 2001. "Computación molecular sin memoria basada en ADN". En XXX (eds.) 2001. *Información: Tratamiento y Representación*. Servicio de Publicaciones, Universidad de Sevilla. Capítulo 1: 3-43.

un experimento realizado en el laboratorio consiguió resolver una instancia concreta de un problema *computacionalmente intratable* usando técnicas de biología molecular para la manipulación del ADN (un problema es computacionalmente intratable si cualquier solución algorítmica conocida del mismo necesita una cantidad de recursos de tipo exponencial en el tamaño del dato de entrada).

Hacia finales de la década de los cincuenta, el premio nobel R.P. Feynman ([10]) describe los ordenadores *sub-microscópicos* e introduce el concepto teórico de *computación a nivel molecular*. Con ello, los organismos vivos, en general, y algunas moléculas en particular, son consideradas potencialmente como máquinas capaces de desarrollar procesos susceptibles de ser interpretados como operaciones de cálculo.

El auge de los actuales ordenadores electrónicos convencionales ha sido posible gracias al invento del *transistor* que sustituyó a las *válvulas y tubos de vacío* de la primera generación de los ordenadores electrónicos. Con los transistores comenzó propiamente la segunda generación (en 1958) y propició, por primera vez, la manipulación electrónica del *silicio*. Los transistores dieron paso, sucesivamente, a los *circuitos integrados* (tercera generación, mediado de la década de los sesenta), compuestos por cientos de transistores, y los *chips* de silicio (cuarta generación, principio de la década de los setenta) que incorporan miles de transistores y basan su funcionamiento en un proceso de grabado a través de un rayo de luz. Así como el transistor permitió por primera vez la manipulación electrónica del *silicio*, el experimento de Adleman puede ser considerado como el primer paso hacia la construcción de un prototipo de ordenador molecular basado en la manipulación bioquímica del *carbono*.

En julio de 2000, un equipo de científicos de la Universidad de California desarrolló un interruptor eléctrico del tamaño de una millonésima de milímetro (un nanómetro), a partir de una molécula. Todo parece indicar que este interruptor puede representar una alternativa revolucionaria en relación con los actuales chips de silicio.

- En su funcionamiento sustituye la luz por una reacción química, lo que representa un importante ahorro en el consumo de energía.
- Estos nuevos interruptores podrían disponer de más de mil procesadores en el espacio ocupado hoy día por un procesador (los actuales chips de silicio tienen una altura aproximada de cinco mil nanómetros).
- Se estima que estos interruptores podrían aumentar la velocidad de procesamiento de la información, cien mil millones de veces la de un ordenador convencional, y podrían reproducir la capacidad equivalente a cien ordenadores convencionales en el tamaño de un grano de sal fina.

Desde los tiempos más remotos, el hombre tuvo el convencimiento de que algunos rasgos de cada generación se transmitían a la siguiente (*todo ser engendra otros semejantes*, sintetizaba una creencia generalizada). No obstante, las reglas y mecanismos que rigen la herencia eran desconocidos por completo. Curiosamente, en 1872, Darwin aseguraba en un artículo (*The expressions of the emotions in man and animals*) que *las leyes que gobiernan la herencia son desconocidas en su mayor parte*, mientras que desde hacía seis años (1866) estaban publicados los trabajos de J.G. Mendel en los *Anales de la Sociedad de Historia Natural de Brunn*. En ellos, Mendel establecía los principios que rigen la herencia (*los caracteres de los padres no se transmiten por azar, sino a través de un mecanismo preciso con entidad de ley*).

A principio del siglo XX se descubrió que los *cromosomas* (descritos por Hofmeister en 1848 y cuya denominación se debe a Waldeyer, 1888) están relacionados directamente con los mecanismos de la herencia, al ser portadores del material genético que determina las características de la descendencia. Entre 1943 (Claude, Porter) y 1947 (Mirsky) se descubre que los cromosomas están compuestos, básicamente, por proteínas y ADN. Debido a la mayor complejidad de la estructura molecular de las proteínas, se tenía el convencimiento de que éstas debían ser las encargadas de transportar la información genética de los padres.

Los trabajos de J. Watson y F. Crick de principio de la década de los cincuenta (entre 1951 y 1953) echaría por tierra el papel relevante atribuido a las proteínas en relación con la herencia. Watson y Crick demuestran que las moléculas de ADN codifican toda la información genética de los organismos vivos, descifran la estructura molecular del ADN, descubren el principio de complementariedad así como el direccionamiento de dichas moléculas, y justifican la posibilidad de usar ciertas técnicas para su manipulación.

En este trabajo se estudia brevemente la estructura del ADN, así como algunas de las operaciones que se pueden realizar en el laboratorio con cadenas del citado ácido. Las secciones 2 y 3 están dedicadas a la descripción y análisis de los experimentos de L. Adleman y de R.J. Lipton que marcan propiamente el comienzo de la computación ADN, con la resolución de sendas instancias concretas de problemas computacionalmente intratables mediante la manipulación en el laboratorio de moléculas de ADN. En la sección 4 se introducen los elementos básicos de un modelo abstracto de computación molecular, se estudian las propiedades relevantes de los modelos sin memoria que usan el ADN como sustrato computacional, y se realiza una propuesta original para describir como sistemas formales, los programas moleculares diseñados en esos modelos para resolver problemas de decisión. Ello permite establecer la verificación formal de programas a través de la demostración de la adecuación (corrección) y completitud de los sistemas formales asociados. En las sec-

ciones 5 y 6 se estudian dos modelos de computación molecular sin memoria basados en ADN: el modelo **no restringido** de L. Adleman ([2]) y el modelo **débil** de M. Amos ([3]). Se ilustra dicho estudio con la presentación de soluciones moleculares de dos problemas **NP**-completos de especial relevancia: el problema de la coloreabilidad de un grafo no dirigido con tres colores, en el modelo no restringido, y el problema del camino hamiltoniano, en su versión dirigida y sin nodos distinguidos, en el modelo débil.

1.1.1 Estructura del ADN

El ácido desoxirribonucleico (ADN) es un polímero que, en su estructura lineal, consta de una serie de monómeros denominados *desoxirribonucleótidos*, a los que llamaremos brevemente *nucleótidos*. A su vez, cada nucleótido consta de:

- (a) Un azúcar (*desoxirribosa*) que tiene cinco átomos de carbono enumerados del 1' al 5', y que en el carbono 3' tiene un grupo hidroxilo (*OH*).
- (b) Un grupo fosfato (*P*), unido al azúcar por el carbono 5'.
- (c) Una base nitrogenada, unida al azúcar por el carbono 1'.

Generalmente se identifica el nucleótido con la base nitrogenada. Existen cuatro tipos de bases nitrogenadas: *adenina*, *citocina*, *guanina* y *timina*, que suelen representarse por las iniciales correspondientes: **A**, **C**, **G**, **T**. La adenina y la guanina pertenecen al grupo de las *purinas*, mientras que la citosina y la timina pertenecen al grupo de las *pirimidinas*.

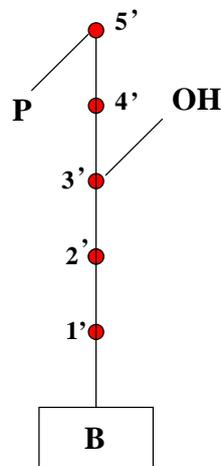


Figura 1. Estructura esquemática de un nucleótido

Los nucleótidos pueden enlazarse de dos maneras diferentes:

- Mediante un *enlace fosfodiéster* (covalente): el grupo fosfato 5' de un nucleótido se une al grupo hidroxilo 3' de otro.
- Mediante un *enlace de hidrógeno* que se realiza a través de las bases nitrogenadas.

El enlace fosfodiéster entre nucleótidos permite elaborar *cadena simple* de ADN que poseen dos extremos de comportamiento muy diferentes tanto química como biológicamente. El tipo de grupo químico de los extremos que están disponibles para nuevos enlaces dan una *polaridad* o *direccionamiento* a las moléculas de ADN; así hablaremos de la dirección 5' → 3' o la dirección 3' → 5'. A la hora de trabajar con cadenas simples, si no se explicita su dirección se sobreentenderá que ésta es 5' → 3'. Si γ es una cadena simple, notaremos $\bar{\gamma}$ la cadena formada por los nucleótidos complementarios (las cadenas complementarias se expresarán, en cambio, en la dirección 3' → 5', salvo que se explicita lo contrario).

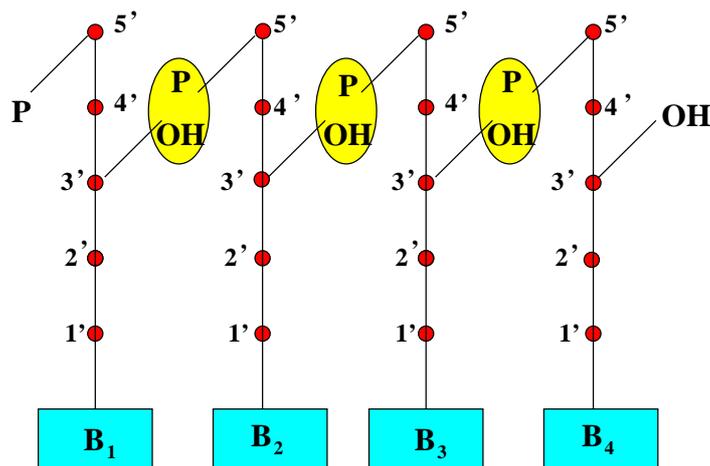


Figura 2. Enlace fosfodiéster: cadena simple 5'-B₁B₂B₃B₄

El enlace de hidrógeno es más débil que el fosfodiéster y se rige por el principio de complementariedad: la adenina sólo se puede enlazar con la timina (y recíprocamente), y la citosina sólo con la guanina (y recíprocamente). Más aún, el enlace entre la adenina y la timina se establece a través de dos puentes de hidrógeno mientras que entre la citosina y la guanina se producen

tres puentes de hidrógeno, lo cual proporciona un poco de más fuerza a este último enlace.

Combinando enlaces fosfodiéster y enlaces de hidrógeno se obtienen *cadena doble* de ADN; es decir, *doble hélice* que se disponen espacialmente formando la estructura conocida como *doble hélice*: dos cadenas simples están alineadas de forma *antiparalela* (es decir, una con la dirección $5' \rightarrow 3'$ y la otra con dirección $3' \rightarrow 5'$) y unidas por enlaces de hidrógeno; los nucleótidos unidos por enlaces fosfodiéster, con los grupos fosfatos orientados hacia el exterior de la hélice y las bases nitrogenadas próximas al centro, lo que proporciona una gran estabilidad a la molécula.

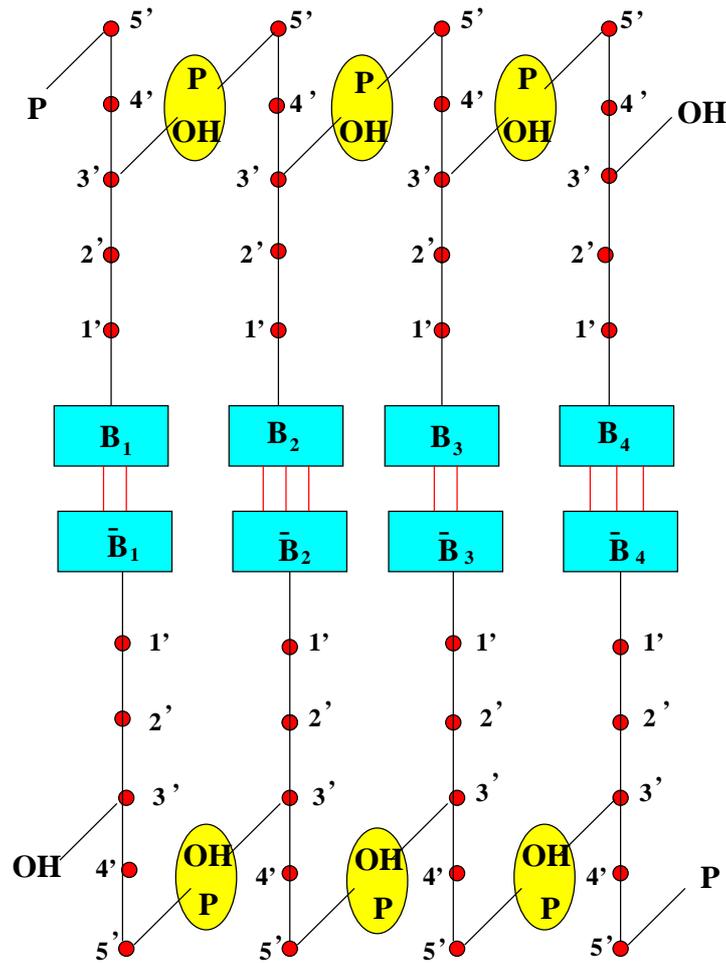


Figura 3. Enlace fosfodiéster + Enlace de hidrógeno: cadena doble

1.1.2 Operaciones con cadenas de ADN

A continuación describimos brevemente algunas operaciones con cadenas de ADN que se pueden realizar actualmente en el laboratorio, y que son la base tanto de la ingeniería genética como de la computación ADN. Para mayor detalle se remite al capítulo 1 de [17] y de [25].

1.1.2.1 Desnaturalización y Renaturalización

Toda doble hebra se puede separar en dos cadenas simples sin que se rompan los enlaces fosfodiéster, para ello basta calentar la solución hasta una temperatura comprendida entre 85 y 95 grados centígrados. Este proceso se denomina de desnaturalización (*melting* o *reannealing*).

También es posible formar una doble hebra a partir de dos cadenas simples complementarias: para ello basta someter la solución a un enfriado lento hasta aproximadamente los 55 grados centígrados, permitiendo que se vayan enlazando las bases nitrogenadas complementarias. Este proceso se denomina de renaturalización (*annealing*).

1.1.2.2 Medida de la longitud de una molécula

La longitud de una cadena simple se define como el número de bases nitrogenadas que contiene, y se expresa en *mer*. La longitud de una cadena doble se define como el número de pares de bases nitrogenadas complementarias que contiene y se expresa en *pares de bases*, bp.

La longitud de una molécula de ADN se puede medir a través de la técnica denominada *electroforesis en gel*. Toda molécula de ADN está dotada de una carga eléctrica negativa que es proporcional a su longitud. Asimismo, la fuerza necesaria para desplazar una molécula también es proporcional a su longitud. Por tanto, si un conjunto de moléculas de ADN se somete a un campo eléctrico en una solución ideal, entonces todas las moléculas se desplazarán hacia el electrodo positivo pero a igual velocidad, con independencia de la longitud que posea. Pues bien, se utilizará un gel a fin de discriminar las moléculas según su longitud.

- En un recipiente rectangular que contiene disolvente, se introduce un cierto gel en polvo y se calienta el contenido para conseguir la disolución del gel.
- A continuación se somete la solución a un proceso de enfriado lento. En el transcurso del mismo se coloca en uno de los extremos del rectángulo una especie de peine, de tal manera que en el momento en que se consiga una cierta consistencia en la solución, finaliza el enfriado y se retira el peine.

- En las distintas ranuras formadas por el peine se colocan las moléculas de ADN que se quieren medir.
- Se activa un campo eléctrico que tiene el electrodo positivo en el lado opuesto a las ranuras.
- El campo se desactiva cuando la primera molécula llega al electrodo positivo.
- La longitud de cada molécula se calcula conociendo la distancia recorrida durante el proceso y el tiempo transcurrido.

1.1.2.3 Extracción

Dado un tubo de ensayo, T , que contiene una solución con cadenas simples de ADN, es posible seleccionar del mismo todas aquellas moléculas que contienen como subcadena, una cierta cadena prefijada, γ . Ello se puede conseguir mediante el *método de las sondas metálicas*, que procede de la siguiente manera:

- Se consideran unas microesferas de hierro que tienen adheridas la cadena $3' - \bar{\gamma}$, y se introducen en el tubo T .
- Se somete la solución a un proceso de renaturalización.
- Se coloca un imán a un lado del tubo, T , y se vierte en otro recipiente, T_1 , las moléculas no *adheridas* a $3' - \bar{\gamma}$.
- Se retira el imán, se añade nuevo disolvente y se somete la solución a un proceso de desnaturalización.
- Se vuelve a colocar un imán a un lado del tubo, T , y se vierte en otro recipiente, T_2 , las moléculas no *adheridas* a $3' - \bar{\gamma}$.
- El tubo T_1 estará formado por las moléculas del tubo inicial que **no** contienen a γ como subcadena, y T_2 por aquellas que **sí** contienen a γ como subcadena.

1.1.2.4 Alargar y copiar una cadena de ADN

La *polimerasa* es una enzima que permite lo que podríamos entender como *reproducción* de las moléculas de ADN. En cierto sentido podemos considerar que esta enzima implementa la complementariedad de Watson–Crick. La polimerasa se desliza sobre una cadena simple, va leyendo los nucleótidos y escribiendo de acuerdo con una cierta ley. Conviene resaltar la similitud entre la acción de esta enzima y la ejecución de una máquina de Turing: la cabeza de trabajo de la máquina se desliza a través de la cinta, va leyendo las casillas

y escribe sobre la misma según le dicta la función de transición de la máquina, que también indica cómo continúa desplazándose la cabeza de trabajo sobre la cinta.

Para copiar una cadena simple se necesita: (a) un *molde* o *patrón* que es una cadena simple que codifica por complementariedad la cadena a copiar; (b) un *cebador* (*primer*) o cadena corta que está enrollada al molde por un extremo; y (c) nucleótidos suficientes en la solución. La polimerasa se activa cuando llega al extremo 3' del cebador y, generalmente, copia en la dirección $5' \rightarrow 3'$.

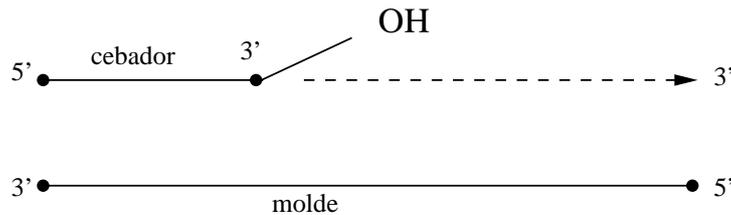


Figura 4. Acción de la polimerasa

1.1.2.5 Síntesis de una cadena de ADN

Esta operación permite fabricar cadenas (simples o dobles) de ADN a partir de una secuencia prefijada. La síntesis de una cadena simple se realiza en la dirección $3' \rightarrow 5'$. El primer nucleótido se adhiere a un soporte sólido por el extremo 3' y se van añadiendo nucleótidos por el extremo 5'. Una cadena corta sintética de nucleótidos recibe el nombre de *oligonucleótido*, o más brevemente, *oligo*. Para sintetizar una doble hebra, en primer lugar se sintetiza la hebra $3' \rightarrow 5'$ y se considera un cebador; después se usa la polimerasa para completar la doble hebra. Estos procesos son automatizables.

1.1.2.6 Acortar cadenas de ADN

Las enzimas *nucleasas* destruyen enlaces fosfodiéster (covalente). La *exonucleasa* es una enzima de este tipo que elimina nucleótidos de una doble hebra, a la vez uno de cada extremo y en la dirección $3' \rightarrow 5'$. Las enzimas nucleasas son más flexibles que la polimerasa: algunas variedades sólo eliminan nucleótidos del extremo 5' y otras sólo del extremo 3'.

1.1.2.7 Cortar cadenas de ADN

Las enzimas *endonucleasas* pueden cortar cadenas (simples o dobles) de ADN por cualquier sitio. No obstante, existe una variedad (las *endonucleasas de restricción*) que sólo pueden cortar doble hebras y por un sitio específico (denominado *lugar de reconocimiento*). Este tipo de enzimas están especializadas según el lugar de reconocimiento y la forma de realizar el corte. Si existen

varios lugares de reconocimiento en una doble hebra, entonces la endonucleasa de restricción realizará un corte por cada uno de ellos.

1.1.2.8 Empastar cadenas de ADN

Al cortar una doble hebra se producen una serie de voladizos que, en determinadas condiciones, pueden volver a enlazarse entre sí. En este caso, así como en otros procesos, pueden aparecer unos pequeños huecos entre nucleótidos consecutivos de una cadena de ADN. La enzima *ligasa* permite rellenar, empastar, dichas hendiduras restaurando la estabilidad y fortaleza de los enlaces fosfodiéster.

1.1.2.9 Alteración de nucleótidos en una cadena de ADN

Existen unas enzimas, denominadas de *modificación*, que alteran ciertas cadenas de ADN añadiendo o quitando unas componentes químicas de las mismas. Este tipo de enzimas suele actuar en colaboración con una enzima de restricción que tiene asociado el mismo lugar de reconocimiento. Las enzimas de modificación son útiles para controlar algunas operaciones con ADN, especialmente aquellas que utilizan un organismo vivo para protegerse de ataques externos; por ejemplo, para defenderse de un cierto virus, una determinada bacteria cuenta con una enzima de restricción cuyo objetivo es destruir cadenas de ADN del virus; dicha enzima trabaja en colaboración con una de modificación, que tiene el mismo lugar de reconocimiento, y cuya misión consiste en alterar los lugares correspondientes en las cadenas de ADN de la bacteria a fin de que ésta no quede afectada por la acción de su propia enzima de restricción.

1.1.2.10 Copias múltiples (amplificación) de cadenas de ADN

La técnica de la *reacción en cadena de la polimerasa* (PCR) fue descubierta por K. Mullis en 1985 y permite generar gran cantidad de copias de una cadena molde o patrón. Esta técnica es muy eficiente y tiene una gran cantidad de aplicaciones en ingeniería genética, análisis del genoma, diagnóstico clínica, arqueología, paleontología, etc.

La técnica de la reacción en cadena de la polimerasa funciona de la siguiente manera:

- Supongamos que queremos fabricar copias de una doble hebra α cuyos bordes $3'$ son β (“borde superior”) y γ (“borde inferior”).
- En la *fase inicial* se prepara una solución que contiene la molécula α , oligos $3' - \bar{\beta}$, $5' - \bar{\gamma}$, nucleótidos en cantidades elevadas, y la enzima polimerasa (se usará una variedad que sea resistente a altas temperaturas).

- La *fase de ejecución* propiamente dicha consta de una serie de ciclos cada uno de los cuales comprende tres pasos:
 - (a) *Desnaturalización*: se calienta la solución hasta una temperatura cercana a la de ebullición para romper los enlaces de hidrógeno.
 - (b) *Renaturalización*: mediante un enfriado lento se producen enlaces entre bordes y oligos.
 - (c) *Extensión*: se calienta nuevamente la solución hasta unos 72 grados para que la acción de la polimerasa extienda los bordes.
- Cada ciclo de la reacción duplica el número de doble hebras α ; por tanto, al reiterar este proceso n veces se obtendrán 2^n copias de la molécula patrón.

1.1.2.11 Lectura de una cadena de ADN

El Proyecto Genoma Humano, iniciado en 1990 y coordinado por instancias públicas de EEUU, tenía como objetivo descifrar antes del año 2003 la secuenciación correcta del ADN humano. Craig Venter, un científico heterodoxo que participaba en dicho Proyecto, creó en 1998 una empresa privada (PE Celera Genomics) en la que consiguió descifrar, en el año 2000, los 3.120 millones de pares de nucleótidos que componen el ADN humano (aún quedan por transcribir unos 40.000 “huecos” y verificar varias veces los resultados de Venter, a fin de elevar a definitivo los resultados provisionales obtenidos). El trabajo consistió en la lectura correcta de una cadena de ADN extraordinariamente grande. Veamos a continuación cómo se puede realizar la lectura de una cadena simple de ADN.

Para ello se consideran ciertos nucleótidos que han sido modificados levemente, denominados *nucleótidos análogos*, que se denotan por ddA , ddC , ddG , ddT , y están asociados a los nucleótidos A , C , G , T , respectivamente. La operación de lectura de una cadena simple, α , de ADN procede como sigue:

- Se extiende α por el extremo $3'$ de acuerdo con un molde γ formando una cadena simple β .
- Se preparan cuatro tubos (T_A, T_C, T_G, T_T) conteniendo cadenas β , oligos $5' - \bar{\gamma}$, polimerasa y nucleótidos. Además, en el tubo T_X se añadirán nucleótidos análogos del tipo ddX , para cada $X \in \{A, C, G, T\}$.
- A partir de la cadena simple β y de los oligos $5' - \bar{\gamma}$, en cada tubo T_X se obtienen doble hebras parciales β' .
- En el tubo T_X , la polimerasa completa las doble hebras parciales β' . No obstante, en este proceso aparecerán doble hebras parciales no completas cuando se elija el nucleótido análogo ddX en lugar de X .

- Se somete la solución a un proceso de desnaturalización y se extrae de la solución las cadenas simples que contienen a $5' - \overline{\gamma}$.
- Se colocan las cadenas extraídas en una solución y se ordenan según su longitud, mediante la técnica de la electroforesis en gel.
- Se leen los nucleótidos del extremo $3'$ de las cadenas ordenadas: la cadena complementaria será α .

1.2 El experimento de Adleman

En noviembre de 1994, L. Adleman ([1]), matemático y doctor en Informática por la Universidad de California, Berkeley, resolvió una instancia concreta de un problema NP-completo (el *problema del circuito hamiltoniano*, en su versión dirigida y con un par de nodos distinguidos [11]) a través de la manipulación de moléculas de ADN usando técnicas de biología molecular.

Problema del circuito hamiltoniano en su versión dirigida y con un par de nodos distinguidos: Sean $G = (V, E)$ un grafo dirigido y $v_i, v_f \in V$ dos nodos distinguidos. Determinar si existe un camino simple desde v_i a v_f que pase por todos los nodos del grafo (un tal camino se dice que es *hamiltoniano* en G).

El experimento de Adleman resolvió el problema citado para el grafo concreto de tamaño siete que aparece en la siguiente figura.

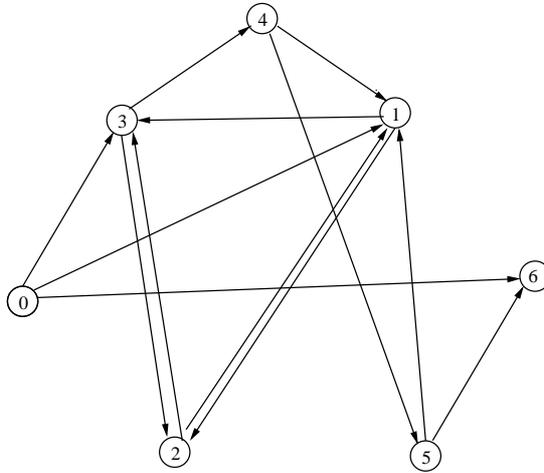


Figura 5. Grafo usado en el experimento de Adleman

La instancia concreta de esta figura posee un único camino hamiltoniano ($0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$), que puede ser detectado por una persona en unos segundos y por un ordenador en menos de una millonésima de segundo, mientras que el experimento de Adleman necesitó siete días para su realización. No obstante, la importancia del citado experimento se puede resumir en los siguientes puntos:

- Proporciona un primer ejemplo de computación a nivel molecular, que potencialmente es un tamaño que nunca podrá ser alcanzado por la industria de los semiconductores.
- Muestra nuevas perspectivas de las moléculas de ADN como estructura de datos peculiares, debido a la complementariedad de Watson-Crick y a la extraordinaria densidad de información capaz de almacenar.
- Ilustra la posibilidad de usar moléculas de ADN para resolver instancias de problemas de tipo combinatorio computacionalmente intratables.
- Manifiesta la capacidad del ADN para simular computaciones de forma masivamente paralela, gracias a la acción simultánea de las enzimas sobre cadenas de ADN.

El experimento de Adleman puede considerarse como la implementación en el laboratorio del siguiente algoritmo de búsqueda exhaustiva que resuelve el problema del camino hamiltoniano citado:

Entrada: $G = (V, E)$, grafo dirigido; v_i y v_f nodos distinguidos.

Paso 1: Generar todos los caminos de G .

Paso 2: Rechazar los caminos que no empiezan por v_i y terminan en v_f .

Paso 3: Rechazar los caminos que no contienen exactamente $|V|$ nodos.

Paso 4: Para cada $u \in V$, rechazar los caminos que no contienen al nodo u .

Salida: **SI**, en caso que quede algún camino; **NO** en caso contrario.

Grosso modo, el experimento de Adleman procede como sigue: a partir de un tubo de ensayo inicial que contiene cadenas de ADN codificando todos los posibles caminos del grafo, se van seleccionando moléculas de acuerdo con los criterios indicados en cada uno de los pasos del algoritmo. Al final de ese proceso, las moléculas de ADN que permanezcan en el tubo de salida codificarán caminos hamiltonianos del grafo que van desde el nodo v_i hasta el nodo v_f .

1.2.1 Implementación de los pasos del algoritmo

La implementación del paso 1 (*generar todos los caminos de G*) consiste propiamente en elaborar un tubo de ensayo inicial que contenga moléculas que codifican cualquier posible camino del grafo considerado. Para ello, a cada nodo i ($0 \leq i \leq 6$) del grafo se le asocia un oligo, s_i , de longitud 20 mer. Para cada i se designa por s'_i (respectivamente, s''_i) los diez primeros (respectivamente, últimos) caracteres de s_i .

A cada arco (i, j) del grafo se le asocia el siguiente oligo, e_{ij} :

$$e_{ij} = \begin{cases} s''_i s'_j & \text{si } i \neq 0 \wedge j \neq 6 \\ s_0 s'_j & \text{si } i = 0 \wedge j \neq 6 \\ s''_i s_6 & \text{si } i \neq 0 \wedge j = 6 \\ s_0 s_6 & \text{si } i = 0 \wedge j = 6 \end{cases}$$

Se parte de un tubo de ensayo que contiene una cierta cantidad de disolvente. Se le añade a la solución una cierta cantidad (50 pmoles) de oligos \overline{s}_i , para cada nodo i ($0 \leq i \leq 6$), y la misma de oligos e_{ij} , para cada arco $(i, j) \in E$. La solución se somete a un proceso de renaturalización, usando la enzima ligasa para empastar las hendiduras que pudieran aparecer entre nucleótidos. De esta manera, se obtiene una serie de doble hebras que codifican caminos del grafo. Más aún, las cantidades iniciales consideradas garantizan con creces que cada camino del grafo aparecerá codificado en el tubo inicial así elaborado.

Para implementar el paso 2 (*rechazar todos los caminos que no empiezan por 0 y terminan en 6*, siendo 0 y 6 los nodos distinguidos) se aplica la técnica PCR utilizando como bordes los oligos s_0 y \overline{s}_6 , respectivamente. Con ello sólo serán amplificados los caminos del grafo que comienzan por 0 y terminan en 6.

A continuación, se utiliza la técnica de la electroforesis en gel para seleccionar las moléculas de longitud 140 bp, a fin de implementar el paso 3 (*rechazar todos los caminos que no contienen exactamente siete vértices*).

El paso 4 (*para cada nodo i del grafo, rechazar todos los caminos que no contienen dicho nodo*) se implementa como sigue: para cada i ($1 \leq i \leq 5$) se seleccionan las moléculas que contienen al oligo s_i , utilizando para ello el procedimiento de las sondas metálicas (obsérvese que por el paso 2 no es necesario considerar aquí los casos $i = 0$ e $i = 6$).

Finalmente, para detectar si hay algún camino en el tubo de ensayo resultante, se amplifica el producto obtenido mediante la técnica PCR y se hace circular las moléculas obtenidas a través de un gel.

1.2.2 Consideraciones acerca del experimento de Adleman

En relación con el experimento de Adleman conviene resaltar las observaciones siguientes:

- Las operaciones moleculares son ejecutadas de manera simultánea sobre todas las moléculas del tubo de ensayo.
- En el tubo de ensayo inicial existe un número de cadenas que es *exponencial* en el tamaño del dato de entrada.
- El número de operaciones moleculares realizadas es *lineal* en el tamaño del dato de entrada.
- La elección aleatoria de oligos de longitud 20 para codificar nodos y arcos (con la excepción de los arcos de extremos 0 y/ó 6), así como las cantidades de cada oligo que se añaden a la solución, se realiza para garantizar que en el tubo de ensayo inicial estén cadenas que codifiquen cualquier camino del grafo (y sólo esos). En el experimento de Adleman la cantidad considerada fue excesiva y aparecían codificados todos los caminos en cantidades elevadas. Desde luego, la longitud seleccionada para los oligos no tiene porqué ser igual para todos ellos y, en general, dependerá de la densidad y del tamaño del grafo.
- A lo largo de la ejecución del experimento aparecen errores debido a que las operaciones moleculares no son *perfectas*: en algún paso se puede seleccionar cadenas no deseadas, o bien rechazar alguna cadena que codifique una solución correcta del problema. No obstante, estos errores pueden ser controlados, en cierta medida, a fin de conseguir los efectos deseados con una “alta probabilidad” ([2]).
- D. Boneh, C. Dunworth y R. Lipton ([6]) conjeturan que el número máximo de moléculas de ADN que se puede procesar en un experimento molecular es del orden de 10^{21} (aproximadamente 2^{70}). En consecuencia, el experimento de Adleman sólo podría ser simulado para grafos de tamaño menor o igual que 70. Como simple curiosidad baste decir que para simular el experimento de Adleman para un grafo de densidad media y tamaño 200, se necesitaría un volumen molecular equivalente al peso de la tierra.
- Algunas de las ventajas que potencialmente proporciona el experimento de Adleman en relación con el mejor supercomputador del momento son las siguientes:
 - ★ *Velocidad de cálculo*: La máxima velocidad de cálculo alcanzada en el experimento fue de $1'2 \times 10^{18}$ operaciones por segundo (en la

cincuentaava parte de una cucharadita de solución, se formaron más de 100 billones de caminos en un segundo), mientras que el mejor supercomputador de la actualidad alcanza un máximo de 10^{12} operaciones por segundo.

- ★ *Consumo de energía:* En el experimento, con 1 julio de trabajo se consiguió realizar hasta 2×10^{19} operaciones (de acuerdo con la segunda ley de la termodinámica, el número máximo de operaciones *irreversibles* por julio es de 34×10^{19}), mientras que el mejor supercomputador de la actualidad es capaz de realizar como máximo 10^9 operaciones por julio.
- ★ *Densidad de almacenamiento de información:* En el experimento se consiguió una densidad de almacenamiento de información de 1 bit por nanómetro cúbico, mientras que el mejor supercomputador de la actualidad tiene una densidad máxima de información del orden de 1 bit por 10^{12} nm³ (un gramo de ADN ocupa en seco 1 centímetro cúbico, aproximadamente, y es capaz de almacenar la información equivalente a más de 1 billón de CD convencionales).

1.2.3 Formulación abstracta

Consideremos una estructura de datos cuyos objetos son los *tubos* de ensayo, entendiendo como tal un multiconjunto finito de cadenas del alfabeto $\Sigma_{ADN} = \{A, C, G, T\}$. Consideremos las siguientes operaciones moleculares:

1. **Extraer** (T, γ): Dado un tubo, T , y una cadena, γ , sobre Σ_{ADN} , esta operación devuelve los tubos $+(T, \gamma)$ y $-(T, \gamma)$ cuyos elementos son, respectivamente, las moléculas de T que contienen (respectivamente, **no** contienen) a γ como subcadena.
2. **Inicial** (T, γ): Dado un tubo, T , y una cadena, γ , devuelve un tubo cuyos elementos son las moléculas de T que “comienzan” por la cadena γ .
3. **Final** (T, γ): Dado un tubo, T , y una cadena, γ , devuelve un tubo cuyos elementos son las moléculas de T que “terminan” con la cadena γ .
4. **Long** (T, n): Dado un tubo, T , y un número natural, $n \geq 1$, devuelve un tubo cuyos elementos son las moléculas de T que tienen longitud n .
5. **Detectar** (T): Dado un tubo, T , devuelve 1 si existe alguna cadena de ADN en T , y devuelve 0 en caso contrario.

Con la estructura de datos considerada y las operaciones moleculares descritas, el experimento de Adleman se puede expresar algorítmicamente como sigue:

```

Entrada:  $T$  (tubo de ensayo inicial)
 $T \leftarrow \text{Inicial}(T, s_0)$ 
 $T \leftarrow \text{Final}(T, s_6)$ 
 $T \leftarrow \text{Long}(T, 140)$ 
para  $i \leftarrow 1$  hasta 5 hacer
     $T \leftarrow +(T, s_i)$ 
Detectar ( $T$ )

```

Conviene resaltar el hecho de que el experimento de Adleman no puede considerarse como un esquema algorítmico molecular, en el siguiente sentido: el tubo de ensayo inicial del experimento sólo se puede utilizar para el grafo concreto para el que fue diseñado; si se considera otro grafo, aunque tenga tamaño siete, habrá que elaborar nuevamente el tubo de ensayo inicial. No obstante, la formulación abstracta antes descrita nos permite elaborar un primer esbozo de programa molecular para resolver el problema del camino hamiltoniano para un grafo dirigido, G , de tamaño $n + 1$ con dos nodos distinguidos v_i y v_f , siguiendo las ideas de Adleman (es decir, codificando nodos y arcos por oligos de longitud l). Si s_0 es el oligo que codifica el nodo v_i y s_n es el oligo que codifica a v_f , entonces se obtiene el siguiente “programa molecular”:

```

Entrada:  $(T, s_0, s_n)$ 
 $T \leftarrow \text{Inicial}(T, s_0)$ 
 $T \leftarrow \text{Final}(T, s_n)$ 
 $T \leftarrow \text{Long}(T, (n + 1) \cdot l)$ 
para  $i \leftarrow 1$  hasta  $n - 1$  hacer
     $T \leftarrow +(T, s_i)$ 
Detectar ( $T$ )

```

1.3 El experimento de Lipton

En abril de 1995, R.J. Lipton ([14]) resolvió una instancia concreta del problema **SAT** de la satisfactibilidad de la *Lógica Proposicional* siguiendo las ideas de Adleman, con la peculiaridad de que el tubo de ensayo inicial no depende del dato de entrada concreto sino únicamente de su “tamaño” (número de variables de la fórmula). De esta manera se tiene una solución molecular de cualquier instancia del problema **SAT** de tamaño prefijado y, en consecuencia, proporciona realmente el primer esquema algorítmico molecular.

1.3.1 El problema de la satisfactibilidad de la Lógica Proposicional

El lenguaje de la Lógica Proposicional, **LP**, consta de:

- (a) Un conjunto infinito numerable, **VP**, de variables (que denominaremos proposicionales).
- (b) Dos conectivas lógicas: \neg (negación) y \vee (disyunción).
- (c) Dos símbolos auxiliares: “(” y “)”.

A partir de la negación y disyunción, se definen de manera natural las conectivas lógicas \wedge (conjunción), \longrightarrow (implicación) y \longleftrightarrow (doble implicación).

El conjunto **PForm** de las fórmulas proposicionales se define recursivamente (mediante un procedimiento generativo) como sigue: es el menor conjunto Γ que verifica, simultáneamente, las condiciones siguientes:

- (a) $\mathbf{VP} \subseteq \Gamma$.
- (b) Es cerrado bajo negación (es decir, si $\varphi \in \Gamma$ entonces $\neg\varphi \in \Gamma$).
- (c) Es cerrado bajo disyunción (es decir, si $\varphi, \psi \in \Gamma$ entonces $\varphi \vee \psi \in \Gamma$).

Un *literal* es una variable proposicional o la negación de una variable proposicional. Una *cláusula* es una disyunción de literales. Se dice que una fórmula proposicional está en *forma normal conjuntiva* si es una conjunción de cláusulas; es decir, si es una conjunción de una disyunción de literales.

Una *valoración* o *asignación de verdad* es una aplicación de **VP** en $\{0, 1\}$. Toda valoración se extiende unívocamente de **VP** al conjunto **PForm** de manera natural (de acuerdo con las clásicas *tablas de verdad* de las conectivas lógicas).

Diremos que una valoración, σ , es *relevante* para una fórmula proposicional, φ , si y sólo si $\sigma(x) = 0$, para cada $x \in \mathbf{VP} - \text{Var}(\varphi)$, siendo $\text{Var}(\varphi)$ el conjunto de las variables proposicionales de φ . Obsérvese que el número de valoraciones relevantes para una fórmula proposicional φ es $2^{|\text{Var}(\varphi)|}$.

Una fórmula proposicional se dice que es *satisfactible* si existe, al menos, una valoración que asigna 1 a dicha fórmula.

Problema de la satisfactibilidad de la Lógica Proposicional:
Dada una fórmula proposicional en forma normal conjuntiva, determinar si es satisfactible.

Como es bien sabido, el problema de la satisfactibilidad de la Lógica Proposicional es **NP**-completo ([11]).

1.3.2 Preparación del experimento

Sea $\varphi \equiv c_1 \wedge \dots \wedge c_p$ con $c_i = l_{i,1} \vee \dots \vee l_{i,r_i}$, una fórmula proposicional en forma normal conjuntiva, cuyo conjunto de variables es $\text{Var}(\varphi) = \{x_1, \dots, x_n\}$. Asociamos a la fórmula φ un grafo dirigido, $G_n = (V_n, E_n)$, definido como sigue:

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

El grafo G_n , descrito en la figura 6, verifica las siguientes propiedades:

- Existen 2^n caminos simples desde a_1 hasta a_{n+1} en G_n .
- Existe una biyección natural entre el conjunto de los caminos anteriormente citados y las valoraciones relevantes para φ : si $\gamma = a_1 x_1^{j_1} a_2 x_2^{j_2} \dots x_n^{j_n} a_{n+1}$ es un camino desde a_1 hasta a_{n+1} , entonces se le asocia la valoración, σ , relevante para φ , caracterizada por las relaciones $\sigma(x_i) = j_i$ ($1 \leq i \leq n$).

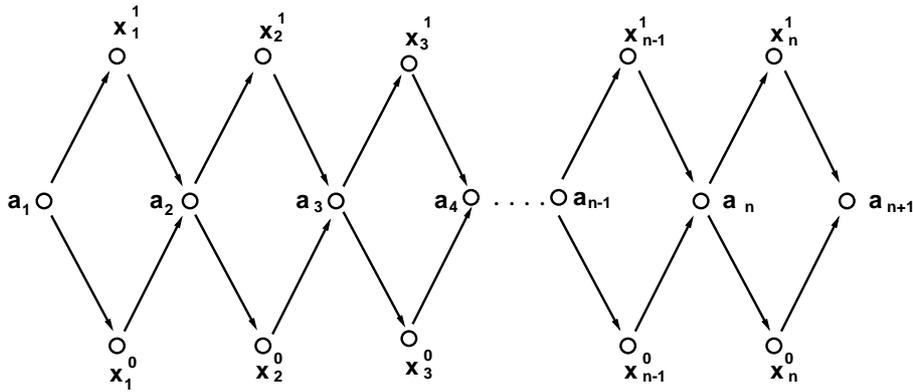


Figura 6. Grafo dirigido asociado a una fórmula proposicional con n variables

Para elaborar el tubo de ensayo inicial, T_0 , se procede de manera similar a la seguida en el experimento de Adleman. A cada nodo, $i \in V_n$, del grafo se le asocia un oligo, s_i , de longitud 20 mer (en realidad este valor dependerá de n). Para cada i se designa por s_i' (respectivamente, s_i'') los diez primeros (respectivamente, últimos) caracteres de la cadena s_i . A cada arco, $(i, j) \in E_n$, del grafo se le asocia el siguiente oligo, $e_{ij} = 3' - \bar{s}_i'' \bar{s}_j'$.

Se parte de un tubo de ensayo que contiene una cierta cantidad de disolvente. Se le añade a la solución una cierta cantidad de oligos s_i , para cada nodo $i \in V_n$, la misma de oligos e_{ij} , para cada arco $(i, j) \in E_n$, y la misma

de oligos $3' - \bar{s}'_{a_1}$ y $3' - \bar{s}''_{a_{n+1}}$. La solución se somete a un proceso de renaturalización, usando la enzima ligasa para empastar los huecos que pudieran aparecer entre nucleótidos. Finalmente se seleccionan las cadenas simples con bordes $5' - s_{a_1}$, para lo cual basta someter la solución a un proceso de desnaturalización y aplicar la operación *extraer* (T_0, s_{a_1}) , que se ha descrito en el experimento de Adleman.

De esta manera, se obtiene una serie de doble hebras que codifican caminos del grafo desde a_1 hasta a_{n+1} . Las cantidades iniciales de oligos que se consideran deben garantizar que cada camino desde a_1 hasta a_{n+1} (y sólo esos) estén codificados en el tubo inicial. Además, debido a la simetría del grafo auxiliar considerado todos los caminos aparecen repetidos el mismo número de veces, aproximadamente. Por tanto, en el tubo de ensayo inicial se obtiene un multiconjunto de moléculas que codifican todas las valoraciones relevantes para la fórmula φ .

1.3.3 Diseño del programa molecular

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

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

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

La idea del experimento de Lipton es la siguiente: a partir del tubo inicial, T_0 , que 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 la cláusula c_1 . Para ello:
 - ★ 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 la cláusula c_2 (así dichas valoraciones hacen verdadera la fórmula $c_1 \wedge c_2$). Para ello:

- ★ 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 $c_1 \wedge \dots \wedge c_p$.

Estas ideas sugieren el diseño del siguiente programa molecular que resuelve el problema de la satisfactibilidad de la Lógica Proposicional:

```

Entrada:  $T_0$ 
para  $i \leftarrow 1$  hasta  $p$  hacer
   $T_1 \leftarrow T_0$ ;  $T_0 \leftarrow \emptyset$ 
  para  $j \leftarrow 1$  hasta  $r_i$  hacer
     $T' \leftarrow +(T_1, l_{i,j}^1)$ 
     $T_1 \leftarrow -(T_1, l_{i,j}^1)$ 
     $T_0 \leftarrow T_0 \cup T'$ 
Detectar( $T_0$ )

```

En donde $T_0 \cup T'$ indica la unión de los tubos T_0 y T' como multiconjuntos, y las operaciones extraer y detectar funcionan tal como se han descrito en la formulación abstracta del experimento de Adleman.

El número de operaciones moleculares que se ejecutan en el programa es lineal en el número de literales, k , de la fórmula de entrada (en concreto, k operaciones de extracción, k operaciones de unión y 1 operación de detección). Además, el número total de tubos usados es

$$1 + \sum_{i=1}^p (3 + 3 \cdot r_i) = 1 + 3p + 3k \in O(k)$$

1.3.4 Extensión del experimento de Lipton

En el experimento de Lipton se proporciona una solución molecular del problema de la satisfactibilidad de fórmulas proposicionales en forma normal conjuntiva. Veamos seguidamente cómo es posible generalizar dicho experimento a fin de decidir la satisfactibilidad de una fórmula proposicional arbitraria.

Proposición 3.1. *Sea φ una fórmula proposicional con k conectivas lógicas. Sea T un tubo que contiene moléculas que codifican valoraciones*

relevantes para φ . Entonces, con $O(k)$ operaciones de extracción, unión y amplificación, se pueden generar los tubos siguientes:

$$T^1(T, \varphi) = \{\sigma \in T : \sigma(\varphi) = 1\} \quad \text{y} \quad T^0(T, \varphi) = \{\sigma \in T : \sigma(\varphi) = 0\}$$

Demostración: Por inducción fuerte sobre el número de conectivas lógicas, k , de la fórmula.

- Si $\varphi \equiv x_i$ y T es un tubo que contiene codificaciones relevantes para φ , entonces $T^1(T, \varphi) = +(T, x_i^1)$ y $T^0(T, \varphi) = -(T, x_i^1)$.

Si $\varphi \equiv \bar{x}_i$ y T es un tubo que contiene codificaciones relevantes para φ , entonces $T^1(T, \varphi) = +(T, x_i^0)$ y $T^0(T, \varphi) = -(T, x_i^0)$.

- Supongamos cierto el resultado para cada $r \leq k$ y sea φ una fórmula con $k + 1$ conectivas lógicas. Sea T un tubo que codifica valoraciones relevantes para φ .

Si $\varphi \equiv \neg\varphi_1$, entonces por hipótesis de inducción con $O(k)$ operaciones de extracción, unión y amplificación, se pueden obtener los tubos

$$\begin{cases} T^1(T, \varphi_1) = \{\sigma \in T : \sigma(\varphi_1) = 1\} \\ T^0(T, \varphi_1) = \{\sigma \in T : \sigma(\varphi_1) = 0\} \end{cases}$$

Entonces basta devolver los tubos

$$\begin{cases} T^1(T, \varphi) = T^0(T, \varphi_1) \\ T^0(T, \varphi) = T^1(T, \varphi_1) \end{cases}$$

Sea $\varphi \equiv \varphi_1 \vee \varphi_2$. Sea s_i el número de conectivas lógicas de φ_i . Por hipótesis de inducción, con $O(s_1)$ operaciones de extracción, unión y amplificación se pueden obtener los tubos

$$\begin{cases} T^1(T, \varphi_1) = \{\sigma \in T : \sigma(\varphi_1) = 1\} \\ T^0(T, \varphi_1) = \{\sigma \in T : \sigma(\varphi_1) = 0\} \end{cases}$$

Amplificamos el tubo $T^0(T, \varphi_1)$ obteniendo dos copias. Entonces, por hipótesis de inducción, con $O(s_2)$ operaciones de extracción, unión y amplificación obtenemos los tubos $T^1(T^0(T, \varphi_1), \varphi_2)$ y $T^0(T^0(T, \varphi_1), \varphi_2)$. En tal situación, basta devolver los tubos

$$\begin{cases} T^1(T, \varphi_1 \vee \varphi_2) = T^1(T, \varphi_1) \cup T^1(T^0(T, \varphi_1), \varphi_2) \\ T^0(T, \varphi_1 \vee \varphi_2) = T^0(T^0(T, \varphi_1), \varphi_2) \end{cases}$$

Si $\varphi \equiv \varphi_1 \wedge \varphi_2$, entonces el razonamiento es análogo al realizado en el caso de la disyunción. ■

Corolario 3.2. *El problema de la satisfactibilidad para fórmulas proposicionales se puede resolver realizando una operación de detectar y $O(k)$ operaciones de extracción, unión y amplificación (en donde k es el número de conectivas lógicas de la fórmula).*

Demostración: Dada una fórmula proposicional con n variables y k conectivas lógicas procedemos, al igual que en la preparación del experimento de Lipton, diseñando el grafo dirigido asociado, G_n , y elaborando, a partir de éste, el tubo de ensayo inicial, T_0 , que contiene cadenas de ADN que codifican todas las valoraciones relevantes para la fórmula de entrada. De la proposición anterior resulta que realizando $O(k)$ operaciones de extracción, unión y amplificación se obtienen los tubos

$$\begin{cases} T^1(T_0, \varphi) = \{\sigma \in T_0 : \sigma(\varphi) = 1\} \\ T^0(T_0, \varphi) = \{\sigma \in T_0 : \sigma(\varphi) = 0\} \end{cases}$$

Entonces basta aplicar la operación *detectar* ($T^1(T_0, \varphi)$). ■

1.4 Modelos moleculares sin memoria

El experimento de Lipton resolvió una instancia concreta del problema de la satisfactibilidad de la Lógica Proposicional y proporcionó, por primera vez, un esquema algorítmico molecular. En esta sección se presenta un marco formal en el que el experimento citado se puede considerar como la ejecución de un procedimiento *mecánico*.

Recordemos que un modelo de computación, M , está caracterizado básicamente por:

- (a) una estructura de datos, \mathcal{D}_M ;
- (b) un conjunto de programas, \mathcal{P}_M ; y
- (c) una función semántica, \mathcal{S}_M , que asigna al par ordenado (P, n) , la función de aridad $n \geq 1$ que *calcula* el programa P .

Los datos de un modelo de computación molecular van a ser unos objetos, denominados *tubos*, asociados a un alfabeto arbitrario prefijado, Σ . Los elementos de cada tubo codificarán moléculas de ADN, con tal de asociar a cada símbolo del alfabeto un oligo verificando ciertas condiciones.

Las instrucciones básicas de un modelo de computación molecular pueden ser de dos tipos:

- *Moleculares*: operaciones basadas en propiedades de las moléculas de ADN.
- *Robóticas*: operaciones secuenciales estándar (bucles, condicionales, asignaciones, etc.) que, básicamente, proporcionan la secuenciación de las operaciones moleculares. Estas operaciones no actúan directamente sobre la estructura de las moléculas de ADN.

La función semántica del modelo se define de manera natural: (a) para las instrucciones moleculares, el resultado correspondiente de la operación molecular que representa; (b) para las instrucciones robóticas, la semántica secuencial convencional.

La elección de las operaciones moleculares primitivas de los modelos que se estudian se debe, entre otros motivos, al hecho de que todas ellas son implementables hoy día en el laboratorio con las técnicas actuales de biología molecular.

En un modelo de computación molecular, conviene distinguir entre lo que es el aparato formal del mismo y lo que representa la implementación práctica que se puede realizar en el laboratorio. Denominaremos *sustrato computacional* de un modelo a la sustancia molecular sobre la que se materializa la implementación del mismo.

A partir de las instrucciones moleculares y de las convencionales, se obtienen de manera natural los programas sobre el modelo. Las entradas de los mismos serán tubos y las salidas serán **SI**, **NO**, o bien un elemento, una molécula, del tubo final (a veces, la salida puede ser el propio tubo final).

El modelo no restringido de L. Adleman y el débil de M. Amos, que se estudiarán en las próximas secciones, son modelos de computación molecular basados en el procedimiento de *filtrado*; es decir, toda computación en dichos modelos comienza con un tubo de ensayo inicial que contiene todas las posibles soluciones del problema y, mediante una serie de procedimientos de selección que produce sucesivos filtrados, devuelve un tubo de salida que contiene todas las soluciones correctas del mismo (y sólo esas). Así pues, las computaciones en dichos modelos no alteran la estructura interna de las moléculas de ADN que componen los tubos, simplemente las moléculas son rechazadas o no, de acuerdo con un determinado test. Por tanto, podemos interpretar que estos modelos de computación carecen de memoria de acceso aleatorio.

A la hora de trabajar en estos modelos hay que distinguir claramente una *fase de iniciación*, en la que se elabora el tubo de ensayo inicial (se ha de garantizar que en ese tubo estén todas las moléculas que codifican posibles soluciones del problema), y una *fase de ejecución*, en la que se van filtrando del tubo inicial las moléculas que codifican soluciones correctas del problema (y sólo esas).

1.4.1 Programas moleculares como sistemas formales

Para verificar formalmente un programa molecular diseñado para resolver un problema, hay que demostrar dos resultados básicos:

- (a) Toda molécula del tubo de salida representa una solución correcta del problema (*corrección del programa molecular*); es decir, si el tubo de salida no está vacío, entonces existe alguna solución correcta del problema; con otras palabras, para un problema de decisión, si el programa devuelve SI, entonces la respuesta del problema es SI.
- (b) Toda molécula del tubo inicial que codifica una solución correcta del problema debe estar en el tubo de salida (*completitud del programa molecular*); es decir, si el tubo de salida está vacío, entonces no existe ninguna solución correcta del problema; con otras palabras, para un problema de decisión, si el programa devuelve NO, entonces la respuesta del problema es NO.

A continuación, vamos a describir como sistemas formales, los programas diseñados para resolver problemas de decisión; de tal manera que demostrar la verificación de un programa molecular sea equivalente a establecer la adecuación y completitud del sistema formal asociado.

Un *sistema formal* consta, en esencia, de

- (a) una *sintaxis*, obtenida a partir de una alfabeto prefijado y de una regla de formación de fórmulas del sistema;
- (b) una *semántica* que proporciona el concepto de *validez*; y
- (c) una *deducción* que proporciona el concepto de demostrabilidad de una fórmula en el sistema.

Se dice que un sistema formal es *adecuado* si y sólo si toda fórmula demostrable es válida, y se dice que es *completo* si y sólo si toda fórmula válida es demostrable.

Un *problema de decisión*, X , es una aplicación de un cierto conjunto $E_X \subseteq \Sigma_X$ en $\{0, 1\}$. Con otras palabras, informalmente, se dice que un problema es de decisión si únicamente admite como respuesta SI o NO. Los elementos de E_X se denominan ejemplares o datos de entrada del problema, y Σ_X es el alfabeto de entrada del mismo.

Sea P un programa molecular diseñado para resolver un problema de decisión, X . Sea Σ_P el alfabeto sobre el que se describe P . Para cada $a \in E_X$, notaremos $I(P, a)$ al conjunto de tubos iniciales de P relativos al dato de entrada a . Para cada $a \in E_X$ y cada $T \in I(P, a)$, notaremos $P(a, T)$ el resultado de ejecutar el programa molecular P con dato de entrada a y tubo de ensayo inicial T .

Definición: Sea P un programa molecular diseñado para resolver un problema de decisión, X . Verificar (X, P) consiste en demostrar que el programa molecular P resuelve el problema de decisión X .

Es decir, verificar (X, P) consiste en probar que para cada $a \in E_X$ y cada $T \in I(P, a)$ se tiene que $P(a, T) = 1$ si y sólo si $X(a) = 1$. En esta equivalencia, la implicación directa recibe el nombre de *corrección* del programa P para el problema X , y la implicación recíproca el de *completitud* del programa P para el problema X .

Definición: Sea P un programa molecular diseñado para resolver un problema de decisión, X . Diremos que (X, P) está verificado si y sólo si para cada $a \in E_X$ y cada $T \in I(P, a)$ se tiene que $P(a, T) = 1$ si y sólo si $X(a) = 1$.

Definición: Sea P un programa molecular diseñado para resolver un problema de decisión, X . El sistema formal, $\mathcal{S}(X, P)$, asociado a (X, P) consta de:

- (a) Un alfabeto, $\Sigma = \Sigma_X \cup \Sigma_P$.
- (b) Un conjunto de fórmulas: una fórmula es un par ordenado (a, T) , en donde $a \in E_X$ y $T \in I(P, a)$.
- (c) Un concepto de validez: una fórmula (a, T) es válida si y sólo si $X(a) = 1$.
- (d) Un concepto de deducción: una fórmula (a, T) es demostrable si y sólo si $P(a, T) = 1$.

Proposición 4.1. Sea P un programa molecular diseñado para resolver un problema de decisión, X . Se verifica:

1. El sistema $\mathcal{S}(X, P)$ es adecuado si y sólo si

$$\forall a \in E_X \forall T \in I(P, a) (P(a, T) = 1 \implies X(a) = 1)$$

2. El sistema $\mathcal{S}(X, P)$ es completo si y sólo si

$$\forall a \in E_X \forall T \in I(P, a) (X(a) = 1 \implies P(a, T) = 1)$$

Corolario 4.2. Sea P un programa molecular diseñado para resolver un problema de decisión, X . Entonces (X, P) está verificado si y sólo si el sistema formal asociado, $\mathcal{S}(X, P)$, es adecuado y completo.

1.5 Modelo no restringido de Adleman

Para introducir un modelo de computación molecular basta definir el concepto de tubo y explicitar las operaciones moleculares primitivas del mismo.

Definición: *Un tubo en el modelo no restringido es un multiconjunto finito de cadenas del alfabeto $\Sigma_{ADN} = \{A, C, G, T\}$.*

Las cadenas del alfabeto Σ_{ADN} pueden ser implementadas de manera natural en el laboratorio a través de moléculas de ADN, gracias al direccionamiento de las mismas. En los modelos moleculares que tienen al ADN como sustrato computacional, el alfabeto habitual es Σ_{ADN} . No obstante, con frecuencia y por cuestiones técnicas, usaremos como alfabeto de esos modelos otro distinto, de tal manera que cada símbolo del nuevo alfabeto está codificado por un determinado oligo en Σ_{ADN}^* (conjunto de todas las cadenas del alfabeto Σ_{ADN}).

Las instrucciones moleculares básicas del modelo no restringido son las siguientes:

- **Extraer**(T, γ): dado un tubo, T , y una cadena, γ , de Σ_{ADN} , devuelve dos tubos:

$$\begin{cases} +(T, \gamma) = \{\{\sigma \in T : \gamma \text{ es subcadena de } \sigma\}\} \\ -(T, \gamma) = \{\{\sigma \in T : \gamma \text{ no es subcadena de } \sigma\}\} \end{cases}$$

En esta expresión, la doble llave indica que se trata de un multiconjunto en lugar de un conjunto.

- **Mezclar**(T_1, T_2): dados dos tubos, T_1 y T_2 , devuelve un nuevo tubo, $T_1 \cup T_2$, que es la unión de ambos, como multiconjuntos.
- **Amplificar** ($T, \{T_1, T_2\}$): dado un tubo T , devuelve dos tubos, T_1 y T_2 que son copias exactas de T .
- **Detectar**(T): dado un tubo, T , devuelve **SI**, en el caso en que T contenga alguna molécula de ADN, y **NO** en caso contrario.

Obsérvese que en el modelo no restringido de Adleman únicamente la operación molecular **extraer** implementa el paralelismo masivo.

A continuación vamos a ilustrar cómo se trabaja en este modelo, estudiando la resolubilidad en el mismo de un problema **NP**-completo: el problema de la coloreabilidad con tres colores de un grafo no dirigido (problema **3-COL**). Para ello, diseñaremos un programa molecular que resuelve el problema citado y estableceremos la verificación del mismo, a través de la adecuación y completitud del sistema formal asociado.

1.5.1 Solución molecular del problema 3-COL

Definición: Sean $G = (V, E)$, un grafo no dirigido, y $k \geq 1$. Una coloración de G con k colores es una aplicación de V en el conjunto $\{1, \dots, k\}$. Diremos que una coloración, f , de G con k colores es válida si y sólo si $\forall u \forall v (\{u, v\} \in E \Rightarrow f(u) \neq f(v))$.

Enunciado del problema 3-COL: Dado un grafo no dirigido, G , determinar si existe una coloración válida de G con tres colores.

Sea $G = (V, E)$ un grafo no dirigido, con $V = \{1, \dots, n\}$. Consideremos el siguiente alfabeto: $\Sigma = \{(p_i, c_j) : 1 \leq i \leq n \wedge 1 \leq j \leq 3\}$, en donde p_1, \dots, p_n son oligos que codifican, respectivamente, los nodos $1, \dots, n$, y c_1, c_2, c_3 son oligos que codifican los colores 1, 2 y 3 respectivamente. Para cada i ($1 \leq i \leq n$) y cada j ($1 \leq j \leq 3$) notaremos $p_i c_j = (p_i, c_j)$, y lo interpretaremos como sigue: el nodo codificado por p_i está coloreado con el color c_j .

El tubo de entrada, T_0 , es el siguiente:

$$T_0 = \{\{\sigma \in \Sigma^n : \exists x_1 \dots \exists x_n (\sigma = (p_1, x_1)(p_2, x_2) \dots (p_n, x_n))\}\}$$

Es decir, T_0 es un multiconjunto finito de moléculas que codifican cualquier coloración del grafo con tres colores (y sólo esas). El tubo inicial T_0 se puede preparar siguiendo las ideas del experimento de Lipton como sigue:

- Para cada i, j ($1 \leq i \leq n \wedge 1 \leq j \leq 3$) se codifica el símbolo $p_i c_j$ por un oligo q_{ij} de longitud $2t$ (siendo t un número fijo asociado al tamaño del grafo). Notaremos q'_{ij} (respectivamente, q''_{ij}) los t primeros (respectivamente, últimos) caracteres de la cadena q_{ij} .
- Se parte de un tubo de ensayo que contiene una cierta cantidad de disolvente. Se le añade a la solución una cierta cantidad de oligos q_{ij} , para cada i, j ($1 \leq i \leq n \wedge 1 \leq j \leq 3$). Añadimos la misma cantidad de oligos $3' - \bar{q}''_{ij} \bar{q}'_{i+1, l}$, para cada i, j, l ($1 \leq i \leq n-1 \wedge 1 \leq j, l \leq 3$) y la misma cantidad de oligos $3' - \bar{q}'_{1j}$ y $3' - \bar{q}''_{n, l}$ para cada j, l ($1 \leq j, l \leq 3$).
- La solución se somete a un proceso de renaturalización, usando la enzima ligasa para empastar los huecos que pudieran aparecer entre nucleótidos. Finalmente se seleccionan las cadenas simples con bordes $5' - q'_{1, j}$, con $1 \leq j \leq 3$, para lo cual basta someter la solución a un proceso de desnaturalización y aplicar la operación `extraer` ($T_0, q'_{1, j}$) (que se implementa mediante la técnica de las sondas metálicas, usando microesferas de hierro que tienen adheridas $3' - \bar{q}'_{1, j}$).

1.5.1.1 Diseño del programa molecular

Notación: Sea $\sigma = p_1x_1 \dots p_nx_n \in T_0$. Para cada r ($1 \leq r \leq n$) notaremos $(\sigma)_r = x_r$; es decir, $(\sigma)_r$ representa el color asociado al nodo r por la coloración codificada por σ . Para cada $i \in V$, notaremos por $E_i = \{\{i, k_1^i\}, \dots, \{i, k_{t_i}^i\}\}$ al conjunto de aristas de G de extremo inferior i , de tal manera que $i < k_1^i < \dots < k_{t_i}^i$. Obsérvese que puede existir $i \in V$ tal que $E_i = \emptyset$ (por ejemplo, $E_n = \emptyset$).

La idea de un programa molecular en el modelo no restringido que permite resolver el problema **3-COL** es la siguiente: a partir del tubo de ensayo inicial, T_0 , que codifica todas las posibles coloraciones con tres colores del grafo de entrada, se procede como sigue:

- Se seleccionan las moléculas que codifican coloraciones con tres colores que son válidas para el subgrafo inducido por las aristas de extremo inferior j , con $j \leq 1$.
- De éstas se seleccionan las moléculas que codifican coloraciones con tres colores que son válidas para el subgrafo inducido por las aristas de extremo inferior j , con $j \leq 2$.
- Y se reitera el proceso $n - 1$ veces.

Estas ideas sugieren el diseño del siguiente programa molecular en el modelo no restringido que resuelve el problema 3-COL:

```

Entrada:  $T_0$ 
  para  $i \leftarrow 1$  hasta  $n - 1$  hacer
    amplificar  $(T_0, \{T'_0, T_1\})$ 
    amplificar  $(T'_0, \{T_2, T_3\})$ 
    para  $j \leftarrow 1$  hasta 3 hacer
       $T_j \leftarrow +(T_j, p_i c_j)$ 
      para  $l \leftarrow 1$  hasta  $t_i$  hacer
         $T_j \leftarrow -(T_j, p_{k_l^i} c_j)$ 
       $T_0 \leftarrow T_1 \cup T_2$ 
       $T_0 \leftarrow T_0 \cup T_3$ 
  Detectar  $(T_0)$ 

```

El número de operaciones moleculares que realiza el programa es el siguiente: $2(n - 1)$ operaciones de **amplificación**; $3(n - 1) + |E|$ operaciones de **extracción**; $2(n - 1)$ operaciones **mezclas** y 1 operación de **detección**. En total, el número de operaciones moleculares es del orden $\theta(\max\{|V|, |E|\})$; es decir, es lineal en el tamaño del dato de entrada.

1.5.1.2 Verificación formal del programa molecular diseñado

A efectos de establecer la verificación formal del programa, etiquetamos los tubos que se obtienen a lo largo de la ejecución, reescribiendo el programa molecular como sigue:

Entrada: T^0
 para $i \leftarrow 1$ hasta $n - 1$ hacer
 amplificar $(T^{i-1}, \{T_0^{i-1}, T_1^{i-1}\})$
 amplificar $(T_0^{i-1}, \{T_2^{i-1}, T_3^{i-1}\})$
 para $j \leftarrow 1$ hasta 3 hacer
 $\bar{T}_j^i \leftarrow +(T_j^{i-1}, p_i c_j)$
 $r \leftarrow i$; $T_{j,r}^i \leftarrow \bar{T}_j^i$
 para $l \leftarrow 1$ hasta t_i hacer
 $T_{j,k_l^i}^i \leftarrow -(T_{j,r}^i, p_{k_l^i} c_j)$
 $r \leftarrow k_l^i$
 $T^{i,j} \leftarrow T_{j,r}^i$
 $\bar{T}^i \leftarrow T^{i,1} \cup T^{i,2}$
 $T^i \leftarrow \bar{T}^i \cup T^{i,3}$
 Detectar (T^{n-1})

En primer lugar, vamos a justificar que el programa está basado en un procedimiento de filtrado; es decir, vamos a probar que la sucesión de tubos $\{T^i : 0 \leq i \leq n - 1\}$ es decreciente por la relación de inclusión.

Lema 5.1. $\forall i (1 \leq i \leq n - 1 \longrightarrow T^i \subseteq T^{i-1})$.

Demostración: Sea i tal que $1 \leq i \leq n - 1$. Para cada $j (1 \leq j \leq 3)$ se tiene que $T^{i,j} \subseteq \bar{T}_j^i \subseteq T_j^{i-1} = T^{i-1}$. Luego, $T^i = \bigcup_{j=1}^3 T^{i,j} \subseteq T^{i-1}$. ■

A continuación vamos a ver que las moléculas del tubo $T^{i,j}$ codifican coloraciones del grafo que asignan el color j al nodo i .

Lema 5.2. $\forall i \forall j (1 \leq i \leq n - 1 \wedge 1 \leq j \leq 3 \longrightarrow \forall \sigma \in T^{i,j} ((\sigma)_i = j))$.

Demostración: Sean i, j tales que $1 \leq i \leq n - 1$ y $1 \leq j \leq 3$. Sea $\sigma \in T^{i,j}$. Como $T^{i,j} \subseteq \bar{T}_j^i = +(T_j^{i-1}, p_i c_j)$, resulta que $\sigma \in T_j^{i-1}$ y $(\sigma)_i = j$. ■

Veamos seguidamente que toda molécula del tubo $T^{i,j}$ pertenece a los tubos $T_{j,k_1^i}^i, \dots, T_{j,k_{t_i}^i}^i$.

Lema 5.3. Para cada i, j tales que $1 \leq i \leq n-1$ y $1 \leq j \leq 3$ se verifica que $\forall \sigma \in T^{i,j} \forall l (1 \leq l \leq t_i \longrightarrow \sigma \in T_{j,k_l^i}^i)$.

Demostración: Sean i, j tales que $1 \leq i \leq n-1$ y $1 \leq j \leq 3$. Sea $\sigma \in T^{i,j}$. Veamos que $\forall l (1 \leq l \leq t_i \longrightarrow \sigma \in T_{j,k_l^i}^i)$. Por inducción débil acotada descendente sobre l .

Teniendo presente que $T^{i,j} = T_{j,k_{t_i}^i}^i$ y que $\sigma \in T^{i,j}$, resulta que $\sigma \in T_{j,k_{t_i}^i}^i$. Sea $l > 1$ ($l \leq t_i$) y supongamos que $\sigma \in T_{j,k_l^i}^i$. Como $T_{j,k_l^i}^i = -(T_{j,k_{l-1}^i}^i, p_{k_{l-1}^i} c_j)$, se deduce que $\sigma \in T_{j,k_{l-1}^i}^i$. ■

Ahora vamos a probar que las moléculas del tubo $T_{j,k_l^i}^i$ codifican coloraciones del grafo que asignan al nodo i el color j , y un color distinto al nodo k_l^i .

Lema 5.4. Para cada i, j tales que $1 \leq i \leq n-1$ y $1 \leq j \leq 3$ se verifica que $\forall l (1 \leq l \leq t_i \longrightarrow \forall \sigma \in T_{j,k_l^i}^i ((\sigma)_i = j \wedge (\sigma)_{k_l^i} \neq j))$.

Demostración: Sean i, j tales que $1 \leq i \leq n-1$ y $1 \leq j \leq 3$. Veamos que $\forall l (1 \leq l \leq t_i \longrightarrow \forall \sigma \in T_{j,k_l^i}^i ((\sigma)_i = j \wedge (\sigma)_{k_l^i} \neq j))$. Por inducción débil acotada ascendente sobre l .

Sea $\sigma \in T_{j,k_1^i}^i = -(T_{j,i}^i, p_{k_1^i} c_j) = -(\overline{T}_j^i, p_{k_1^i} c_j)$. Por una parte $(\sigma)_{k_1^i} \neq j$. Por otra, $\sigma \in \overline{T}_j^i = +(T_j^{i-1}, p_i c_j)$. Luego, $(\sigma)_i = j$.

Sea $l < t_i$ ($l \geq 1$) y supongamos cierto el resultado para l . Sea $\sigma \in T_{j,k_{l+1}^i}^i = -(T_{j,k_l^i}^i, p_{k_{l+1}^i} c_j)$. Por una parte se tiene que $(\sigma)_{k_{l+1}^i} \neq j$. Por otra, $\sigma \in T_{j,k_l^i}^i$. Luego, de la hipótesis de inducción, resulta que $(\sigma)_i = j$. ■

Para establecer la corrección formal del programa molecular diseñado, consideremos la siguiente fórmula:

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

Es decir, la fórmula $\theta(i)$ expresa que toda molécula del tubo T^i codifica una coloración del grafo con tres colores que es válida para el subgrafo inducido por el conjunto de vértices $\{1, \dots, i\}$.

Teorema 5.5 $\forall i (1 \leq i \leq n-1 \longrightarrow \theta(i))$. Es decir, la fórmula θ es un invariante del bucle principal.

Demostración: Por inducción débil sobre i .

Sean $\sigma \in T^1$ y $s > 1$ tal que $\{1, s\} \in E$. Sea $j \in \{1, 2, 3\}$ tal que $\sigma \in T^{1,j}$. Del lema 3 se deduce que $\sigma \in T_{j,s}^1$, y del lema 4 concluimos que $(\sigma)_1 \neq (\sigma)_s$.

Sea $i < n - 1$ ($i \geq 1$) y supongamos cierto el resultado para i . Sean $\sigma \in T^{i+1}$ y r, s tales que $1 \leq r \leq i + 1 \wedge r < s \wedge \{r, s\} \in E$.

- Si $1 \leq r \leq i$, teniendo presente que $T^{i+1} \subseteq T^i$, resulta que $\sigma \in T^i$. De la hipótesis de inducción se deduce que $(\sigma)_r \neq (\sigma)_s$.
- Si $r = i + 1$, entonces existe $j \in \{1, 2, 3\}$ tal que $\sigma \in T^{i+1, j}$. Para cada s tal que $i + 1 < s$, $\{i + 1, s\} \in E$, del lema 5.3 resulta que $\sigma \in T_{j, s}^{i+1}$, y del lema 5.4 concluimos que $(\sigma)_{i+1} = j \wedge (\sigma)_s \neq j$.

■

Corolario 5.6. (*Corrección del programa*) Toda molécula del tubo de salida codifica una coloración válida del grafo con tres colores.

Demostración: Tras la ejecución del programa molecular, se tiene que la fórmula $\theta(n - 1)$ es verdadera. Luego, para cada $\sigma \in T^{n-1}$ se verifica que $\forall r \forall s (1 \leq r \leq n - 1 \wedge r < s \wedge \{r, s\} \in E \rightarrow (\sigma)_r \neq (\sigma)_s)$. Es decir, la molécula σ codifica una coloración válida del grafo con tres colores.

■

Para establecer la completitud del programa molecular diseñado, consideremos la siguiente fórmula:

$$\delta(i) \equiv \forall \sigma \in T^0 ([\forall r \forall s (1 \leq r \leq n - 1 \wedge r < s \wedge \{r, s\} \in E \rightarrow (\sigma)_r \neq (\sigma)_s)] \rightarrow \sigma \in T^i)$$

Es decir, la fórmula $\delta(i)$ expresa que toda molécula del tubo de ensayo inicial que codifica una coloración válida del grafo con tres colores, pertenece al tubo T^i .

Teorema 5.7. $\forall i (1 \leq i \leq n - 1 \rightarrow \delta(i))$. Es decir, la fórmula δ es un invariante del bucle principal.

Demostración: Por inducción débil sobre i .

Sea $\sigma \in T^0$ tal que

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

Se tiene que $\sigma \in T_{(\sigma)_1}^0$ y, además, $p_1 c_{(\sigma)_1}$ es una subcadena de σ . Luego,

$$\sigma \in +(T_{(\sigma)_1}^0, p_1 c_{(\sigma)_1}) = \overline{T}_{(\sigma)_1}^1 = T_{(\sigma)_1, 1}^1 \subseteq T^{1, (\sigma)_1} \subseteq T^1$$

Sea $i < n - 1$ ($i \geq 1$) y supongamos cierto el resultado para i . Sea $\sigma \in T^0$ tal que codifica una coloración válida del grafo con tres colores. Por hipótesis de inducción se deduce que $\sigma \in T^i$. Se tiene que $\sigma \in T_{(\sigma)_{i+1}}^i$ y, además, $p_{i+1} c_{(\sigma)_{i+1}}$ es una subcadena de σ . Luego,

$$\sigma \in +(T_{(\sigma)_{i+1}}^i, p_{i+1} c_{(\sigma)_{i+1}}) = \overline{T}_{(\sigma)_{i+1}}^{i+1} = T_{(\sigma)_{i+1}, i+1}^{i+1} \subseteq T^{i+1, (\sigma)_{i+1}}$$

Teniendo presente que $T^{i+1, (\sigma)_{i+1}} \subseteq T^{i+1}$, se concluye que $\sigma \in T^{i+1}$.

■

Corolario 5.8. (*Completitud del programa*) *Toda molécula del tubo de ensayo inicial que codifica una coloración válida del grafo con tres colores, está en el tubo de salida.*

Demostración: Basta tener presente que tras la ejecución del programa molecular, la fórmula $\delta(n - 1)$ es verdadera, y que el tubo de salida del programa es T^{n-1} . ■

1.6 Modelo débil de Amos

Definición: *Un tubo en el modelo débil es un multiconjunto finito de cadenas del alfabeto $\Sigma_{ADN} = \{A, C, G, T\}$.*

Recuérdese que, al igual que en todo modelo de computación molecular que use el ADN como sustrato computacional, por cuestiones técnicas usaremos con frecuencia un alfabeto distinto de Σ_{ADN} , de manera que cada símbolo del nuevo alfabeto estará codificado por un determinado oligo en Σ_{ADN}^* . Las instrucciones moleculares primitivas del modelo débil son las siguientes:

- **Quitar**($T, \{\gamma_1, \dots, \gamma_k\}$): Dado un tubo, T , y un número finito de cadenas, $\gamma_1, \dots, \gamma_k$, de Σ , devuelve el tubo obtenido de T eliminando todas aquellas cadenas que contengan, al menos, una ocurrencia de alguna de las cadenas $\gamma_1, \dots, \gamma_k$ (téngase presente que $T = \text{quitar}(T, \emptyset)$).
- **Copiar**($T, \{T_1, \dots, T_k\}$): Dado un tubo, T , y un número natural, $k \geq 2$, devuelve k tubos, T_1, \dots, T_k , que son copias exactas de T .
- **Unión**($\{T_1, \dots, T_k\}, T$): Dados los tubos T_1, \dots, T_k , con $k \geq 2$, devuelve un tubo, T , cuyo contenido es la unión de los tubos T_1, \dots, T_k como multiconjuntos.
- **Selección**(T): Dado un tubo, T , selecciona aleatoriamente un elemento de T en el caso en que $T \neq \emptyset$; en caso contrario, devuelve **NO**.

Obsérvese que en el modelo débil únicamente la operación molecular **quitar** implementa el paralelismo masivo.

A continuación, vamos a ilustrar cómo se trabaja en este modelo estudiando la resolubilidad en el mismo de un problema computacionalmente intratable que no es de decisión: el problema de generar todas las permutaciones de elementos de un conjunto finito. Terminaremos la sección utilizando este resultado para resolver en el modelo débil, el problema del camino hamiltoniano en su versión dirigida y sin nodos distinguidos.

1.6.1 Generación de permutaciones en el modelo débil

Definición: Dado un número natural, $n \geq 1$, una permutación de orden n es una aplicación biyectiva del conjunto finito $\{1, \dots, n\}$ en sí mismo.

Problema: Dado un número natural, $n \geq 2$, generar todas las permutaciones de orden n .

Para resolver este problema en el modelo débil, consideremos el siguiente alfabeto: $\Sigma = \{(p_i, c_j) : 1 \leq i, j \leq n\}$, en donde p_i es un oligo que codificará la posición i -ésima en la permutación, y c_j es un oligo que codificará el número j . Para cada i, j ($1 \leq i, j \leq n$) notaremos $p_i c_j = (p_i, c_j)$, y lo interpretaremos como sigue: el número j está en la posición i -ésima de la permutación.

El tubo de entrada, T_0 , es el siguiente:

$$T_0 = \{\{\sigma \in \Sigma^n : \exists x_1 \dots \exists x_n (\sigma = (p_1, x_1)(p_2, x_2) \dots (p_n, x_n))\}\}$$

Es decir, el tubo de entrada es un multiconjunto finito de moléculas que codifican todas las posibles sucesiones de longitud n de elementos del conjunto $\{1, \dots, n\}$.

Este tubo se puede preparar siguiendo las ideas del experimento de Lipton, como hemos explicitado en la elaboración del tubo de ensayo inicial para el programa molecular que resolvía el problema **3-COL** en el modelo no restringido (basta considerar ahora $\{c_1, \dots, c_n\}$ en lugar de $\{c_1, c_2, c_3\}$).

Notación: Sea $\sigma = p_1 x_1 \dots p_n x_n \in T_0$. Para cada r ($1 \leq r \leq n$) notaremos $(\sigma)_r = x_r$; es decir, $(\sigma)_r$ representa el número que ocupa la posición r -ésima en la sucesión de longitud n codificada por σ .

1.6.1.1 Diseño del programa molecular

La idea de un programa molecular en el modelo débil para generar todas las permutaciones del conjunto $\{1, \dots, n\}$ es la siguiente: a partir del tubo de ensayo inicial, T_0 , que codifica todas las posibles sucesiones de longitud n del conjunto $\{1, \dots, n\}$ se procede como sigue:

- Se seleccionan las moléculas, σ , que codifican sucesiones tales que

$$\forall r > 1 ((\sigma)_1 \neq (\sigma)_r)$$

- De éstas se seleccionan las moléculas, σ , que codifican sucesiones de longitud n del conjunto $\{1, \dots, n\}$ tales que

$$\forall r > 2 ((\sigma)_1 \neq (\sigma)_r \wedge (\sigma)_2 \neq (\sigma)_r)$$

- Y se reitera el proceso $n - 1$ veces.

Estas ideas sugieren el diseño del siguiente programa molecular en el modelo débil que genera todas las permutaciones de orden n :

Entrada: T_0
 para $j \leftarrow 1$ hasta $n - 1$ hacer
 copiar $(T_0, \{T_1, \dots, T_n\})$
 para $i \leftarrow 1$ hasta n hacer
 quitar $(T_i, \{p_{jr} : r \neq i\} \cup \{p_{ki} : j + 1 \leq k \leq n\})$
 unión $(\{T_1, \dots, T_n\}, T_0)$
 Salida: T_0

El número de operaciones moleculares que realiza el programa es el siguiente: $n - 1$ operaciones copiar, $n - 1$ operaciones unión, y $n(n - 1)$ operaciones quitar. En total, el número de operaciones moleculares es cuadrático en n .

1.6.1.2 Verificación formal del programa molecular

A efectos de establecer la verificación formal del programa molecular, etiquetamos los tubos que se obtienen a lo largo de la ejecución, reescribiendo el programa molecular como sigue:

Entrada: T^0
 para $j \leftarrow 1$ hasta $n - 1$ hacer
 copiar $(T^{j-1}, \{T_1^j, \dots, T_n^j\})$
 para $i \leftarrow 1$ hasta n hacer
 $\bar{T}_i^j \leftarrow$ quitar $(T_i^j, \{p_{jr} : r \neq i\} \cup \{p_{ki} : j + 1 \leq k \leq n\})$
 unión $(\{\bar{T}_1^j, \dots, \bar{T}_n^j\}, T^j)$
 Salida: T^{n-1}

Notación: Para cada $\sigma \in T_0$ y cada j ($1 \leq j \leq n$) notaremos por $A_{\sigma,j}$ el conjunto $\{(\sigma)_1, \dots, (\sigma)_j\}$.

Para establecer la corrección del programa molecular diseñado, consideremos la siguiente fórmula:

$$\theta(j) \equiv \forall \sigma \in T^j (|A_{\sigma,j}| = j \wedge \forall r (j + 1 \leq r \leq n \longrightarrow (\sigma)_r \notin A_{\sigma,j}))$$

Es decir, la fórmula $\theta(j)$ expresa que toda molécula del tubo T^j codifica una sucesión de longitud n de $\{1, \dots, n\}$ tal que los j primeros términos son distintos entre sí y, además, distintos de los restantes términos de la sucesión.

Teorema 6.1. $\forall j$ ($1 \leq j \leq n - 1 \longrightarrow \theta(j)$). Es decir, la fórmula θ es un invariante del bucle principal.

Demostración: Por inducción débil sobre j .

Sea $\sigma \in T^1$. Existe $x \in \{1, \dots, n\}$ tal que

$$\sigma \in \overline{T}_x^1 = \text{quitar}(T_x^1, \{p_1 r : r \neq x\} \cup \{p_k x : 2 \leq k \leq n\})$$

Entonces $A_{\sigma,1} = \{(\sigma)_1\} = \{x\}$; es decir, $|A_{\sigma,1}| = 1$. Además, si $2 \leq k \leq n$, entonces $\sigma \in \overline{T}_x^1 \implies (\sigma)_1 = x \wedge (\sigma)_k \neq x \implies (\sigma)_k \notin A_{\sigma,1}$.

Sea $j < n - 1$ ($j \geq 1$) y supongamos que el resultado es cierto para j . Sea $\sigma \in T^{j+1}$. Sea $x \in \{1, \dots, n\}$ tal que

$$\sigma \in \overline{T}_x^{j+1} = \text{quitar}(T_x^{j+1}, \{p_{j+1} r : r \neq x\} \cup \{p_k x : j+2 \leq k \leq n\})$$

Entonces se verifica que

$$\sigma \in T_x^{j+1} = T^j \wedge (\sigma)_{j+1} = x \wedge \forall k (j+2 \leq k \leq n \longrightarrow (\sigma)_k \neq x)$$

De la hipótesis de inducción se tiene que

$$|A_{\sigma,j}| = j \wedge \forall r (j+1 \leq r \leq n \longrightarrow (\sigma)_r \notin A_{\sigma,j})$$

Teniendo presente que $A_{\sigma,j+1} = A_{\sigma,j} \cup \{(\sigma)_{j+1}\}$ y que $(\sigma)_{j+1} \notin A_{\sigma,j}$, resulta que $|A_{\sigma,j+1}| = j+1$. Finalmente si $j+2 \leq k \leq n$, entonces $(\sigma)_k \notin A_{\sigma,j}$ y $(\sigma)_k \neq x = (\sigma)_{j+1}$. Por tanto, $(\sigma)_k \notin A_{\sigma,j+1}$. ■

Corolario 6.2. (*Corrección del programa*) Toda molécula del tubo de salida codifica una permutación de orden n .

Demostración: Sea $\sigma \in T^{n-1}$. Puesto que la fórmula $\theta(n-1)$ es verdadera resulta que $|A_{\sigma,n-1}| = n-1 \wedge \forall r (n-1+1 \leq r \leq n \longrightarrow (\sigma)_r \notin A_{\sigma,n-1})$. Como $A_{\sigma,n} = A_{\sigma,n-1} \cup \{(\sigma)_n\}$, concluimos que $|A_{\sigma,n}| = n$. Por tanto, la molécula σ codifica una permutación de orden n . ■

Para establecer la completitud del programa molecular diseñado, consideremos la siguiente fórmula:

$$\delta(j) \equiv \forall \sigma \in T^0 (|A_{\sigma,n}| = n \longrightarrow \sigma \in T^j)$$

Es decir, la fórmula $\delta(j)$ expresa que toda molécula del tubo T^i codifica una sucesión de longitud n de $\{1, \dots, n\}$ tal que los j primeros términos son distintos entre sí y, además, distintos de los restantes términos de la sucesión.

Teorema 6.3. $\forall j (1 \leq j \leq n-1 \longrightarrow \delta(j))$. Es decir, la fórmula δ es un invariante del bucle principal.

Demostración: Por inducción débil sobre j .

Sea $\sigma \in T^0$ tal que $|A_{\sigma,n}| = n$. Entonces la molécula σ codifica una permutación de orden n . Se tiene que $\sigma \in T_{(\sigma)_1}^1$ y $\forall r (2 \leq r \leq n \longrightarrow (\sigma)_r \neq (\sigma)_1)$.

Luego, $\sigma \in \text{quitar}(T_{(\sigma)_1}^1, \{p_{1r} : r \neq (\sigma)_1\} \cup \{p_k(\sigma)_1 : 2 \leq k \leq n\})$. Por tanto, $\sigma \in \overline{T}_{(\sigma)_1}^1 \subseteq T^1$.

Sea $j < n - 1$ ($j \geq 1$) y supongamos que el resultado es cierto para j . Sea $\sigma \in T^0$ tal que $|A_{\sigma,n}| = n$. De la hipótesis de inducción resulta que $\sigma \in T^j$. Luego, $\sigma \in T_{(\sigma)_{j+1}}^{j+1}$ y $\forall r$ ($j + 2 \leq r \leq n \rightarrow (\sigma)_r \neq (\sigma)_{j+1}$). De donde se deduce que $\sigma \in \overline{T}_{(\sigma)_{j+1}}^{j+1} \subseteq T^{j+1}$. ■

Corolario 6.4. (*Completitud del programa*) *Toda molécula del tubo de ensayo inicial que codifica una permutación de orden n , está en el tubo de salida.*

Demostración: Basta tener presente que tras la ejecución del programa molecular, la fórmula $\delta(n - 1)$ es verdadera, y que el tubo de salida del programa es T^{n-1} . ■

1.6.2 El problema del camino hamiltoniano sin nodos distinguidos

Problema del camino hamiltoniano en su versión dirigida y sin nodos distinguidos: *dado un grafo dirigido, determinar si existe un camino simple que pasa por todos los nodos del grafo (es decir, determinar si el grafo posee un camino hamiltoniano).*

Vamos a dar una solución molecular a este problema en el modelo débil. Sea $G = (V, E)$ un grafo dirigido, con $V = \{1, \dots, n\}$. Consideremos el siguiente alfabeto $\Sigma = \{(p_i, c_j) : 1 \leq i, j \leq n\}$, en donde p_i es un oligo que codificará la posición i -ésima del camino, y c_j es un oligo que codificará el nodo j . Para cada i, j ($1 \leq i, j \leq n$) notaremos $p_i c_j = (p_i, c_j)$, y lo interpretaremos como sigue: el nodo j está en la posición i -ésima del camino.

El tubo de ensayo inicial, T_0 , es un multiconjunto finito de moléculas que codifican todas las posibles permutaciones de orden n . Por tanto, dicho tubo puede ser elaborado como tubo de salida de un programa molecular que genere todas las permutaciones de orden n (por ejemplo, el programa que hemos presentado en 1.6.1.1).

Notación: Si $\sigma = p_1 x_1 p_2 x_2 \dots p_n x_n \in T_0$, entonces para cada r ($1 \leq r \leq n$) notaremos $(\sigma)_r = x_r$ y $\sigma_r = ((\sigma)_1, (\sigma)_2, \dots, (\sigma)_r)$; es decir, $\sigma_r = (x_1, x_2, \dots, x_r)$. Además, interpretaremos la molécula σ de esta manera: si σ codifica un camino del grafo, entonces x_1 será el primer nodo del camino, x_2 el segundo, y así sucesivamente. Téngase presente que la molécula σ codifica una permutación de orden n ; en consecuencia si, además, codifica

un camino del grafo, éste será hamiltoniano.

La idea que nos va a permitir resolver en el modelo débil, el problema del camino hamiltoniano en su versión dirigida y sin nodos distinguidos, es la siguiente: a partir del tubo de ensayo inicial, T_0 , que codifica todas las posibles permutaciones de orden n :

- Se seleccionan las moléculas, σ , tales que σ_2 es un camino de G ;
- De éstas, se seleccionan las moléculas, σ , tales que σ_3 es un camino de G (para lo cual, teniendo presente que σ_2 es un camino de G basta que se verifique la condición $((\sigma)_2, (\sigma)_3) \in E$).
- Y se reitera el proceso $n - 1$ veces.

Estas ideas sugieren el diseño del siguiente programa molecular en el modelo débil que resuelve el problema del camino hamiltoniano citado:

```
Entrada:  $T_0$  (multiconjunto que contiene todas las
           permutaciones de orden  $n$ )
para  $i \leftarrow 1$  hasta  $n - 1$  hacer
   $T_0 \leftarrow$  quitar  $(T_0, \{jp_{i+1}k : (j, k) \notin E\})$ 
  seleccionar( $T_0$ )
```

El programa realiza $n - 1$ operaciones quitar y 1 operación seleccionar. Luego el número de operaciones moleculares es lineal en el tamaño del grafo.

1.6.2.1 Verificación formal del programa molecular

Para establecer la verificación formal del programa, procedemos al etiquetado de los tubos que aparecen a lo largo de la ejecución, reescribiendo el programa molecular como sigue:

```
Entrada:  $T_0$ 
para  $i \leftarrow 1$  hasta  $n - 1$  hacer
   $T_i \leftarrow$  quitar  $(T_{i-1}, \{jp_{i+1}k : (j, k) \notin E\})$ 
  seleccionar( $T_{n-1}$ )
```

Para establecer la corrección del programa molecular diseñado, consideremos la siguiente fórmula:

$$\theta(i) \equiv \forall \sigma \in T_{i-1} \forall r (1 \leq r < i \longrightarrow ((\sigma)_r, (\sigma)_{r+1}) \in E)$$

Es decir, la fórmula $\theta(i)$ expresa que toda molécula, σ , del tubo T_{i-1} verifica que σ_i es un camino del grafo G .

Teorema 6.5. $\forall i (2 \leq i \leq n \longrightarrow \theta(i))$. Es decir, la fórmula θ es un invariante del bucle principal.

Demostración: Por inducción débil sobre i .

Sea $\sigma \in T_1$. Puesto que $T_1 = \text{quitar}(T_0, \{jp_2k : (j, k) \notin E\})$, se deduce que $((\sigma)_1, (\sigma)_2) \in E$.

Sea i tal que $2 \leq i < n$ y supongamos cierto el resultado para i . Sea $\sigma \in T_i$. Teniendo presente que $T_i = \text{quitar}(T_{i-1}, \{jp_{i+1}k : (j, k) \notin E\})$, resulta que $\sigma \in T_{i-1}$ y $((\sigma)_i, (\sigma)_{i+1}) \in E$. De la hipótesis de inducción se deduce que $\forall r (1 \leq r < i \rightarrow ((\sigma)_r, (\sigma)_{r+1}) \in E)$. Por tanto, σ_{i+1} es un camino del grafo G . ■

Corolario 6.6. (*Corrección del programa*) Toda molécula del tubo de salida codifica un camino hamiltoniano del grafo G .

Demostración: Sea $\sigma \in T_{n-1}$. Como la fórmula $\theta(n)$ es verdadera resulta que $\forall r (1 \leq r < n \rightarrow ((\sigma)_r, (\sigma)_{r+1}) \in E)$. Por tanto, σ_n es un camino de G . Teniendo presente que la molécula σ codifica una permutación de orden n , concluimos que σ codifica un camino hamiltoniano del grafo G . ■

Para establecer la completitud del programa molecular diseñado, consideremos la siguiente fórmula

$$\delta(i) \equiv \forall \sigma \in T_0 ([\forall r (1 \leq r < n \rightarrow ((\sigma)_r, (\sigma)_{r+1}) \in E)] \rightarrow \sigma \in T_{i-1})$$

Es decir, la fórmula $\delta(i)$ expresa que toda molécula del tubo inicial que codifica un camino hamiltoniano de G pertenece al tubo T_{i-1} .

Teorema 6.7. $\forall i (2 \leq i \leq n \rightarrow \delta(i))$. Es decir, la fórmula δ es un invariante del bucle principal.

Demostración: Por inducción débil sobre i .

Sea $\sigma \in T_0$ tal que σ codifica un camino hamiltoniano de G . Entonces $\forall r (1 \leq r < n \rightarrow ((\sigma)_r, (\sigma)_{r+1}) \in E)$. Luego,

$$\sigma \in \text{quitar}(T_0, \{jp_2k : (j, k) \notin E\}) = T_1$$

Sea i tal que $2 \leq i < n$ y supongamos cierto el resultado para i . Sea $\sigma \in T_0$ tal que σ codifica un camino hamiltoniano de G . Por hipótesis de inducción se tiene que $\sigma \in T_{i-1}$. Luego, $\sigma \in \text{quitar}(T_{i-1}, \{jp_{i+1}k : (j, k) \notin E\}) = T_i$. ■

Corolario 6.8. (*Completitud del programa*) Toda molécula del tubo de ensayo inicial que codifica un camino hamiltoniano de G , está en el tubo de salida.

Demostración: Basta tener presente que la fórmula $\delta(n)$ es verdadera y que el tubo de salida del programa es T_{n-1} . ■

1.7 Bibliografía

- [1] ADLEMAN, L. Molecular Computation of Solutions to Combinatorial Problems. *Science*, 268, November 1994, 1021–1024.
- [2] ADLEMAN, L. On constructing a molecular computer, in *DNA based computers*, R.J. Lipton and E.B. Baum, eds., American Mathematical Society, 1996, 1–22.
- [3] AMOS, M. *DNA computing*. Ph. D. Thesis, University of Warwick, 1997.
- [4] AMOS, M.; WILSON, S.; HODGSON, D.A.; OWENSON, G.; GIBBONS, A. Practical implementation of DNA computations, in *Unconventional Models of Computation*, Springer, 1998, 1–18.
- [5] BEAVER, D. A universal molecular computer, in *DNA based computers*, R.J. Lipton and E.B. Baum, eds., American Mathematical Society, 1996, 29–36.
- [6] BONEH, D.; DUNWORTH, C.; LIPTON, R.J. Breaking DES using a molecular computing, in *DNA based computers*, R.J. Lipton and E.B. Baum, eds., American Mathematical Society, 1996, 37–66.
- [7] BONEH, D.; DUNWORTH, C.; LIPTON, R.J.; SGALL, J. On the computational power of DNA. *Discrete Applied Mathematics*, 71, 1996, 79–94.
- [8] BOREK, E. *The Code of Life*. Columbia University Press, 1965.
- [9] CALUDE, C.S.; CASTI, J.; DINNEEN, eds. *Unconventional Models of Computation*. Springer, 1998.
- [10] FEYNMAN R.P. There's plenty of room at the bottom, in D.H. Hilbert, ed., *Miniaturization*, Reinhold, 1961, 282–296.
- [11] GAREY M.R.; JOHNSON D.S. *Computers and intractability*. W.H. Freeman and Company, New York, 1979.
- [12] KARI, L. DNA Computing: Arrival of Biological Mathematics. *The Mathematical Intelligencer*, Springer-Verlag, New York, vol. 19, n. 2, 1997, 9–22.
- [13] LANDWEBER, L.F.; BAUM, E.B., eds. *DNA Based Computers II*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, vol. 44, 1998.

- [14] LIPTON R.J. DNA Solution of Hard Computational Problems. *Science*, 268, April 1995, 542–545.
- [15] LIPTON, R.J.; BAUM E.B., eds. *DNA Based Computers*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, vol 27, 1996.
- [16] LUNDH, D.; OLSSON, B.; NARAYANAN, A., eds. *Biocomputing and emergent computation*. World Scientific, 1997.
- [17] PĂUN, GH.; ROZENBERG, GR.; SALOMAA, A. *DNA Computing. New Computing Paradigms*. Springer, 1998.
- [18] PÉREZ-JIMÉNEZ, M.J.; SANCHO, F.; GRACIANI, M.C.; ROMERO, A. NP completitud y Computación ADN. *Actas del Encuentro de Matemáticos Andaluces*, vol 2., 2000, 539–548.
- [19] PÉREZ-JIMÉNEZ, M.J.; SANCHO, F.; GRACIANI, M.C.; ROMERO, A. Soluciones moleculares del problema SAT de la Lógica Proposicional, en *Lógica, Lenguaje e Información, JOLL'2000*, A. Nepomuceno y otros, eds., Ed. Kronos, 2000, 243–252.
- [20] PEVZNER, P.A. *Computational Molecular Biology. An Algorithmic Approach*. The MIT Press, 2000.
- [21] PISANTI, N. *A survey on DNA computing*. Technical report, TR–97–07. Università di Pisa, 1997.
- [22] ROWEIS, S.; WINFREE, E.; BURGOYNE, R.; CHELYAPOV, N.V.; GOODMAN, M.F.; ROTHEMUND, P.W.K.; ADLEMAN, L. A Sticker-Based Model for DNA Computation. *Journal of Computational Biology*, vol. 5, num. 4, 1998, 615–629.
- [23] RUBIN, H.; WOOD, D.H., eds. *DNA Based Computers III*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, vol. 48, 1999.
- [24] SALOMAA, A. *Turing, Watson-Crick and Lindenmayer. Aspects of DNA complementarity*. Turku Centre for Computer Science. Technical report, 128, 1997.
- [25] SINDEN, R.R. *DNA Structure and Function*. Academic Press, 1994.