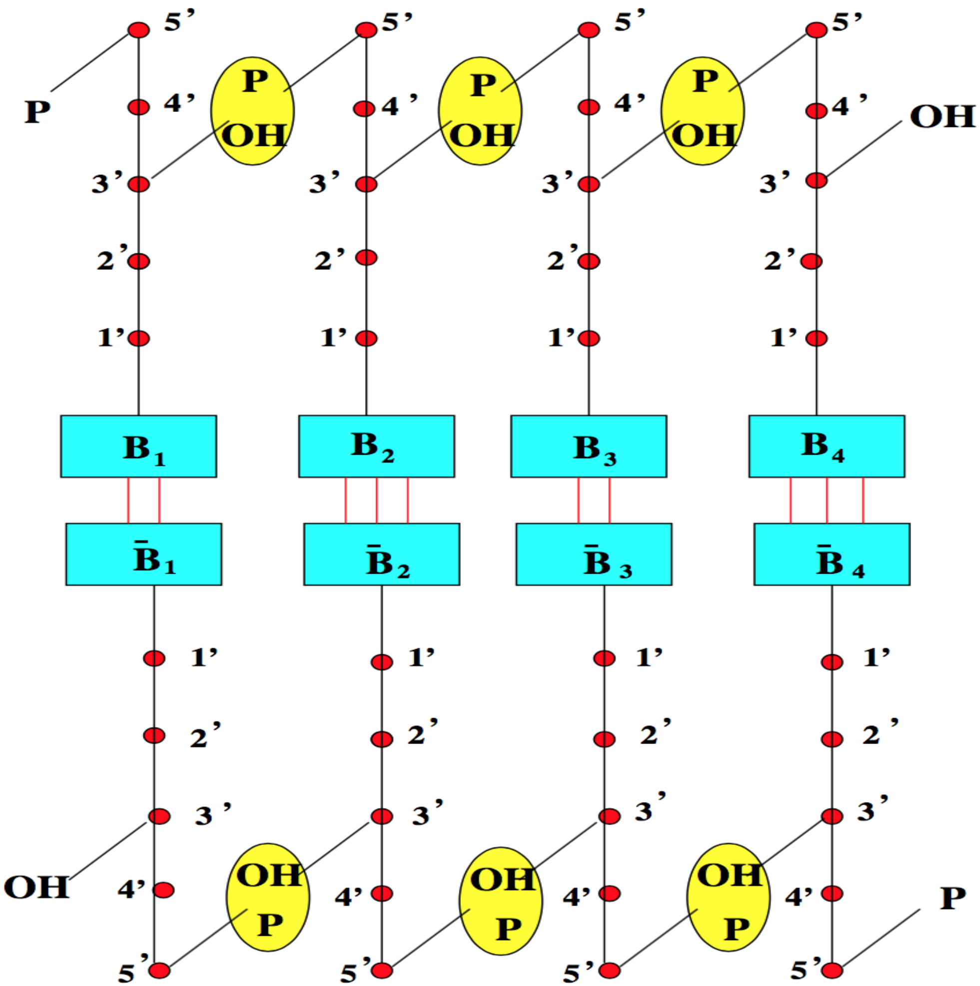
Computación molecular sin memoria basada en ADN

Problemas de generación de permutaciones y camino hamiltoniano

Sergio Rodríguez Calvo, Septiembre 2017.

Computación Bioinspirada (MULCIA), Universidad de Sevilla.

ADN



Computación Molecular

- Tubo de ensayo contiene una solución con cadenas simples de ADN (oligos).
- Automatización de procesos sobre los tubos que realizan operaciones abstractas, tales como, medir, sumar, etc.
- Necesario un modelado y representación del problema adecuado para este tipo de computación.

Operaciones con moléculas de ADN

Algunos ejemplos de operaciones son:

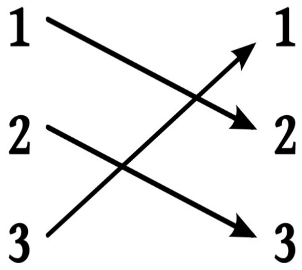
- Desnaturalización: separar doble hebra calentando solución hasta un rango de 85ºC - 95ºC.
- Extracción: extraer de un tubo todas las moléculas que contienen una determinada subcadena, utilizando el método de las sondas metálicas.
- Cortar cadenas: uso de enzimas endonucleasas que cortan cadenas (simples o dobles) por cualquier sitio.

Modelo débil de Amos

- Tubo de ensayo con un multiconjunto finito de cadenas con alfabeto `{A,C,G,T}`.
- Operaciones en el modelo débil de Amos (primitivas) son:
  - `Quitar(T,{s1,...,sn})`.
  - `Copiar(T,{T1,...,Tn})`.
  - `Unión({T1,...,Tn})`.
  - `Selección(T)`.

Problema de la generación de permutaciones

- Permutación:



- Problema: *dado un numero natural n mayor o igual que 2, generar todas las permutaciones de orden n.*

Diseño molecular

- Alfabeto `(pi,cj)` para todo `i,j` entre `[1,n]`.
- Dado un tubo de entrada `T0` que contiene todas las posibles sucesiones:

Entrada: T_0

para $j \leftarrow 1$ **hasta** $n-1$ **hacer**

copiar $(T_0, \{T_1, \ldots, T_n\})$

para $i \leftarrow 1$ **hasta** n **hacer**

quitar $(T_i,\ \{p_j r : \ r \neq i\} \cup \{p_k i : \ j+1 \leq k \leq n\})$

unión $(\{T_1, \ldots, T_n\}, T_0)$

Salida: T_0

Previo a la verificación formal

- Buscar una fórmula que cumpla (p. corrección y completitud):
  - Fórmula es verdadera antes de comenzar el bucle.
  - Fórmula es invariante en dicho bucle (por inducción débil).
- Reetiquetado.
- Corrección del programa (Teorema + Corolario).
- Completitud del programa (Teorema + Corolario).

Verificación formal

- Se reescribe el código para enumerar los tubos:

Entrada: T^0

    para $j \leftarrow 1$ **hasta** $n-1$ **hacer**

        copiar $(T^{j-1}, \{T_1^j, \ldots, T_n^j\})$

        para $i \leftarrow 1$ **hasta** n **hacer**

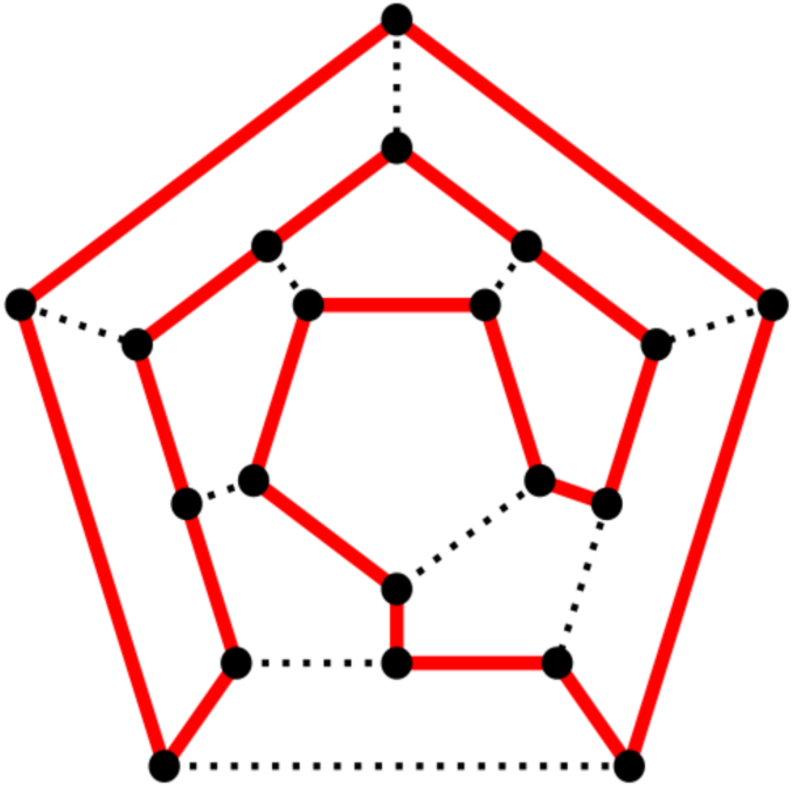
            $\overline{T}_i^j \leftarrow$ **quitar** $(T_i^j, \{p_j r : \; r \neq i\} \cup \{p_k i : \; j+1 \leq k \leq n\})$

        unión $(\{\overline{T}_1^j, \ldots, \overline{T}_n^j\}, T^j)$

Salida: T^{n-1}

- *Ver documento.*

Problema del camino hamiltoniano



Diseño molecular

- Alfabeto `(pi,cj)` para todo `i,j` entre `[1,n]`.
- Dado un tubo de entrada `T0` que contiene todas las posibles sucesiones:

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)

Verificación formal

- Se reescribe el código para enumerar los tubos:

$$
\begin{array}{l}
\texttt{Entrada: } T_0 \\
\quad \texttt{para } i \leftarrow 1 \texttt{ hasta } n-1 \texttt{ hacer} \\
\qquad T_i \leftarrow \texttt{quitar } (T_{i-1}, \{j p_{i+1} k : \ (j,k) \notin E\}) \\
\quad \texttt{seleccionar}(T_{n-1})
\end{array}
$$

- *Ver documento.*

Gracias