



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

FUNCIONES BOOLEANAS A PARTIR DE ADN

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

MATEMÁTICO

P R E S E N T A:

JOSÉ VÍCTOR HERNÁNDEZ URBINA

TUTOR: DR. RICARDO STRAUZ SANTIAGO



FACULTAD DE CIENCIAS
UNAM

2008



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno
Hernández
Urbina
José Víctor
5608-6620
Universidad Nacional Autónoma de México
Facultad de Ciencias
Matemáticas
402079129
2. Datos del tutor
Dr.
Ricardo
Strausz
Santiago
3. Datos del sinodal 1
Dr.
Javier
Bracho
Carpizo
4. Datos del sinodal 2
Dr.
Sergio
Rajsbaum
Gorodezky
5. Datos del sinodal 3
Dr.
Ricardo
Gómez
Aiza
6. Datos del sinodal 4
Dr.
José de Jesús
Galaviz
Casas
7. Datos del trabajo escrito
Funciones booleanas a partir de ADN
86 pp
2008

*A Rosario y a Víctor,
por el amor y las oportunidades.*

*A Pilar,
por todo aquello de lo que se alimenta el corazón.*

*A todos y cada uno de mis profesores y maestros
(que no es lo mismo), y entre todos ellos, a Dino.*

A los pioneros...

Agradecimientos

Hace algunos años, cuando aún me faltaban dos semestres para concluir mis estudios de Ingeniería Cibernética y Sistemas Computacionales en la Universidad La Salle, decidí dejar mi empleo de aquel entonces para dedicarme a una actividad que me satisficiera más que programar tiendas virtuales, aprender las reglas del comercio electrónico y administrar un servidor de correo electrónico; fue entonces que decidí estudiar, como segunda carrera, la licenciatura en Matemáticas en la UNAM. De eso ya suman casi siete años en los cuales gané varios kilos de peso, viajé por Europa, me diagnosticaron asma, volví a tocar la batería, dejé de frecuentar algunos lugares y algunas personas para comenzar a frecuentar *otros* lugares y *otras* personas, y también aprendí *algo* de lo que hacen los *matemáticos*; en la comprensión y adquisición de su pensamiento, que en sentido figurado puede equipararse con la acción de labrar un material, me ayudaron muchas personas, entre profesores y alumnos, tanto de la misma Facultad como de La Salle, y al estar pensando en escribir una sección de agradecimientos me pregunté si uno debe agradecer a aquellas personas por las que se *materializaron* las ideas expuestas en esta tesis, o en general, a toda esa gente que, de una forma u otra, hicieron más sencillo mi paso por la Facultad, desde el día en que pise sus fríos salones por primera vez, hasta el día de hoy; si bien, varias de esas personas actualmente se encuentran persiguiendo sus aspiraciones por otras partes del mundo; así que este agradecimiento probablemente se vuelva tal que varios de sus destinatarios estén ausentes, pero pensándolo dos veces, se hace una labor simbólica al mencionarlos aquí.

En primer lugar quisiera agradecer a Dios: a ese inefable cómputo, hipóstasis del Universo, que dispuso el Azar por el cual surgieron las ideas que la Voluntad transformó en las palabras que llenan las siguientes hojas. En segundo lugar, doy gracias a mis papás por extender sin objeciones la beca *Hernández-Urbina* por el tiempo que duraron mis estudios en la UNAM y por ser siempre la mejor guía en mis decisiones; asimismo quiero agradecer a Pilar Alanís por la inspiración, el apoyo, el interés, por el manual de L^AT_EX y por las excelentes imágenes que ilustran este trabajo. Agradezco también a todos y cada uno de los miembros del jurado por el tiempo que se tomaron en leer mi trabajo y por el empeño en hacer de él un trabajo estético, coherente, comprensible, didáctico e interesante. De igual forma, quiero agradecer a mis hermanos Rodrigo y César, a mis amigos y a la gente con la que llegué a formar incansables equipos de trabajo, en especial a Tzolkin, Roxana, Fal, a los Emilianos, a las Ivonnes, a Iván y a Lizardo que sin su ayuda me las hubiera visto negras en *cuántica*, a mis amigos en La Salle, en especial a los hermanos Rodríguez, a Uriel, Menchaca y Jesús por las pláticas metafísicas. Mando un saludo y agradecimiento a *La Manga del Muerto* y a todos sus *fans*, al Caracho, a Cosmo el perro, a los cafeteros de la Selva de Coyoacan y a mis amigos

en Telefónica, que directa o indirectamente permitieron la redacción de este trabajo, en especial al Robert; y en general, a toda la gente que se interesó por los avances de esta tesis. Por último quiero agradecer a todos mis profesores en la Facultad, en especial al Dr. Ricardo *Dino* Strausz por la motivación, el tiempo y la paciencia, pero sobre todo, por creer en esta tesis.

*Ab Imo Pectore,
JVHU.
Noviembre 2008.*

Índice general

1. Introducción	8
2. Preliminares	13
2.1. La Computación Genética	13
2.2. Operaciones con ADN	14
2.3. El Experimento de Adleman	19
2.4. El Autómata de Benenson-Shapiro: el Doctor ADN	22
2.5. DNA-N-AND	24
3. 1a Aproximación	26
3.1. $NAND_1$ con $n = 2$	27
3.2. $NAND_1$ con $n \geq 2$	37
4. 2a Aproximación	53
4.1. TM <i>in abstracto</i>	54
4.2. TM Molecular	58
5. Conclusiones	78
5.1. Algunas consideraciones acerca de la implementación de las máquinas de ADN	78
5.2. Discusión	81
5.3. Reflexión final	82

Sumario

En las siguientes páginas se presentará una aplicación de la computación genética a las funciones lógicas. Se mostrará el diseño de la compuerta lógica N-AND (negación de la conjunción) a partir de moléculas de ADN.

Se sabe que la función lógica N-AND es *universal* en el sentido de que cualquier otra función lógica puede construirse a partir de esta función, de allí que tenga sentido hablar de la posibilidad de imitar la operación de cualquiera de las funciones *booleanas* (esto es, lógicas) cuando se consigue simular el comportamiento de esta función. En este trabajo se exploran dos versiones distintas de dicha función, que resultan de la manera en que ésta opera sobre las cadenas de entrada. De tal manera que cada una de estas versiones nos remite respectivamente a las nociones de *autómata determinista finito* y *máquina de Turing*, que son objetos matemáticos donde se desarrolla la idea de *computabilidad*.

El comportamiento de ambas máquinas será simulado por la acción de moléculas de material genético. (A su vez, estas máquinas simulan el comportamiento de la compuerta lógica mencionada.) De tal forma que será necesario hacer una abstracción de algunas de las operaciones que se pueden realizar sobre las cadenas de ADN; así como también, realizar una codificación molecular de los símbolos que forman las cadenas de entrada de estas máquinas (los símbolos 0 y 1).

Así, conforme se vaya avanzando en la lectura de esta tesis, se irá evolucionando hacia una notación que prescindir de varias de las características del material genético, conservando tan sólo aquellas que permiten un manejo más abstracto del mismo, y por ende, más matemático; permitiendo así una prueba *rigurosa* de que el diseño de estas máquinas es el correcto, y que éstas computan *efectivamente* la función lógica N-AND.

Capítulo 1

Introducción

El trabajo que se presenta a continuación está motivado por el artículo de Vineet Gupta et al. [1] donde se presenta una técnica para resolver operaciones booleanas (y aritméticas) mediante moléculas de Ácido Desoxirribonucleico (*ADN* en el resto de esta tesis) utilizando resultados recientes sobre este nuevo modelo de cómputo. El problema que emerge en dicho artículo es que los procedimientos utilizados en él (y que deberían de resolver el problema de las operaciones booleanas mediante ADN) carecen de una correcta automatización; es decir, dependen en gran medida del operador, lo cual contradice la esencia de la computación, a saber: la automatización de procesos que al ser humano costarían esfuerzo y/o tiempo en una situación en la que ambos (esfuerzo y tiempo) son de disponibilidad limitada.

Así, la propuesta del presente trabajo es la realización de una máquina (mejor dicho, un par de ellas) basadas en ADN que logran resolver una función booleana específica y de forma automática; es decir, minimizando la intervención del operador.

A continuación se presentan ejemplos de la acción de las máquinas moleculares que se expondrán en esta tesis. La descripción en este punto es sucinta, pero conforme se vaya leyendo, la operación de estas máquinas será más claro.

La función booleana, que se pretende resolver molecularmente es la función *N-AND* (negación de la conjunción lógica), siguiendo una estrategia similar a la expuesta en el artículo citado [1]. Dicha función se puede utilizar para simular cualquier otra función booleana; dicho en otras palabras, cualquier función lógica puede ser escrita en términos de negaciones de conjunciones. La *tabla de verdad* de esta función para dos entradas p y q está descrita en el cuadro (1.1).

Cuadro 1.1: Tabla de Verdad de la función N-AND.

p	q	p NAND q
0	0	1
0	1	1
1	0	1
1	1	0

Nosotros atenderemos 2 aproximaciones a la función N-AND que resultan de la forma en que la función actúa sobre las cadenas binarias de entrada; cadenas de ceros y unos que en la jerga computacional se conocen como *bytes*.

Las aproximaciones son las siguientes:

1a Aproximación. $f : \{0, 1\}^n \rightarrow \{0, 1\}$, $n \geq 2$. Se toma una cadena binaria de longitud n como entrada y la función devuelve un solo *bit*. La forma en que opera aquí la función N-AND es como se ejemplifica a continuación.

- 10011 (primero se realiza la función NAND de 1 y 0, su resultado se pone en el lugar más a la derecha, en este caso, en el de 0)
- 1011 (ahora se toma como entrada el 1 que resultó del cálculo pasado y el siguiente carácter, 0. El resultado se pone en el lugar de 0)
- 111 (ahora se calcula N-AND con entrada 1 y 1, cuyo es 0 se coloca en la posición del carácter más a la derecha)
- 01 (ahora se calcula con el resultado del cálculo anterior y el carácter 1 en la cadena original)
- 1 (aquí el cómputo ha finalizado arrojando como resultado un 1)

Así, se puede decir que el resultado de aplicar la función N-AND a la cadena 10011 es 1.

2a Aproximación. $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, $n \geq 1$ En esta aproximación se toman 2 cadenas (o *bytes*) de la misma longitud y la función devuelve otra cadena de la misma longitud con el resultado del cómputo de la función N-AND entrada por entrada. Ejemplos:

1. 1 N-AND 0 = 0
2. 1001 N-AND 0110 = 1111

Esta aproximación se acerca más a la forma en que los procesadores de las computadoras convencionales realizan las operaciones lógicas y aritméticas más comunes.

Así, en los capítulos siguientes se simularán ambas *Aproximaciones* mediante moléculas de ADN que realizan operaciones muy simples como lo son *cortar* y *pegar* cadenas de ADN. Visto así, en esta tesis se describirá cómo con estas dos simples operaciones puede simularse la acción de un autómata finito determinista y la acción de un determinista aun más complejo: una *Máquina de Turing*; objetos matemáticos empleados en la teoría de la computación que nos servirán para resolver la función N-AND en las acepciones arriba descritas y cuya definición se dará en los siguientes capítulos.

Por ejemplo, implementaremos un autómata determinista finito en una cadena doble de ADN que recibe una cadena de dos entradas binarias (1 y 0), a su vez codificadas en cadenas dobles de ADN, delimitadas por un carácter de terminación (el carácter τ). Asimismo, las transiciones del autómata están implementadas en cadenas dobles de ADN que son agregadas a la solución que contiene la cadena de entrada más otras moléculas que se encargarán de reportar si el autómata *aceptó* o *rechazó* la entrada. Para la operación de la máquina se requieren de dos tipos de enzimas: una cumple la función de cortar la cadena de entrada siguiendo un esquema específico (se trata de la enzima de restricción *FokI* que corta una cadena doble

de ADN toda vez que en ella se encuentre una secuencia particular de nucleótidos); la otra cumple la función inversa, es decir, la de pegar dos cadenas complementarias (es el caso de la enzima *Ligasa* que pega dos cadenas simples cuando son complementarias).

La figura (1.1) muestra el diagrama de estados y transiciones que el autómata debe obedecer. El estado S_3 representa el estado de aceptación de esta máquina operando sobre un alfabeto de 0s y 1s.

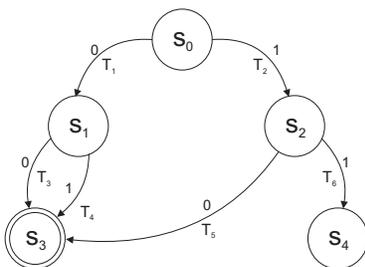


Figura 1.1: Autómata Determinista Finito para simular la función N-AND

La figura (1.2) muestra la manera de construir los símbolos 0, 1 y τ mediante cadenas dobles de ADN. (Esta figura también muestra cómo cada parte de estas cadenas sirve para identificar el estado y entrada actual de la máquina en cualquier punto del cómputo; lo cual será descrito a detalle en los capítulos siguientes.)

Símbolo	0	1	τ
Versión molecular con codificación de <estado, símbolo> en el cabo pegajoso.			

Figura 1.2: Símbolos y su codificación de los estados

Por tanto, podemos construir la cadena de entrada que se muestra en la figura (1.3) la cual consta de las siguientes partes: la cadena de reconocimiento de la enzima de restricción *FokI*; 7 nucleótidos espaciadores, es decir, nucleótidos de composición arbitraria; la entrada codificada como se describe en la figura (1.2) más el carácter delimitador τ ; y por último 512 nucleótidos espaciadores.



Figura 1.3: Ejemplo de una Molécula de Entrada

Las transiciones descritas en la figura (1.1) se codifican en ADN resultando en las moléculas que se muestran en la figura (1.4) y que llamaremos *moléculas de transición*; mientras que las moléculas que nos informan si el autómata se encuentra en estado de aceptación o de rechazo al final del cómputo se denominan *moléculas de detección* y están codificadas como se muestra en la figura (1.5).

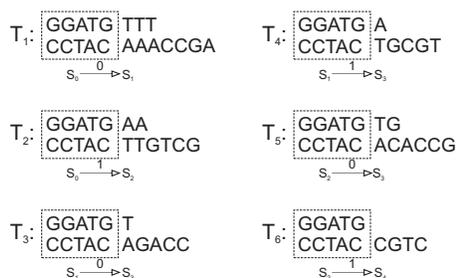


Figura 1.4: Moléculas de Transición



Figura 1.5: Moléculas de Detección

Teniendo en cuenta la codificación de los elementos del autómata anterior (la cadena de entrada, el alfabeto, las transiciones y la manera de detectar el estado final del cómputo) podemos ejemplificar una *corrida* de la máquina en su versión molecular. La figura (1.6) describe paso por paso el cómputo de la máquina molecular para la cadena de entrada 01. Sabemos, siguiendo el diagrama de la figura (1.1), que el autómata acepta esta cadena de entrada, lo cual se interpreta como un 1 de salida, mientras que en el caso en que la cadena sea rechazada (lo cual ocurriría si la entrada fuera la cadena 11) se toma la salida como un 0. En otras palabras, las moléculas de detección, en este caso, hacen las veces de símbolo de salida.

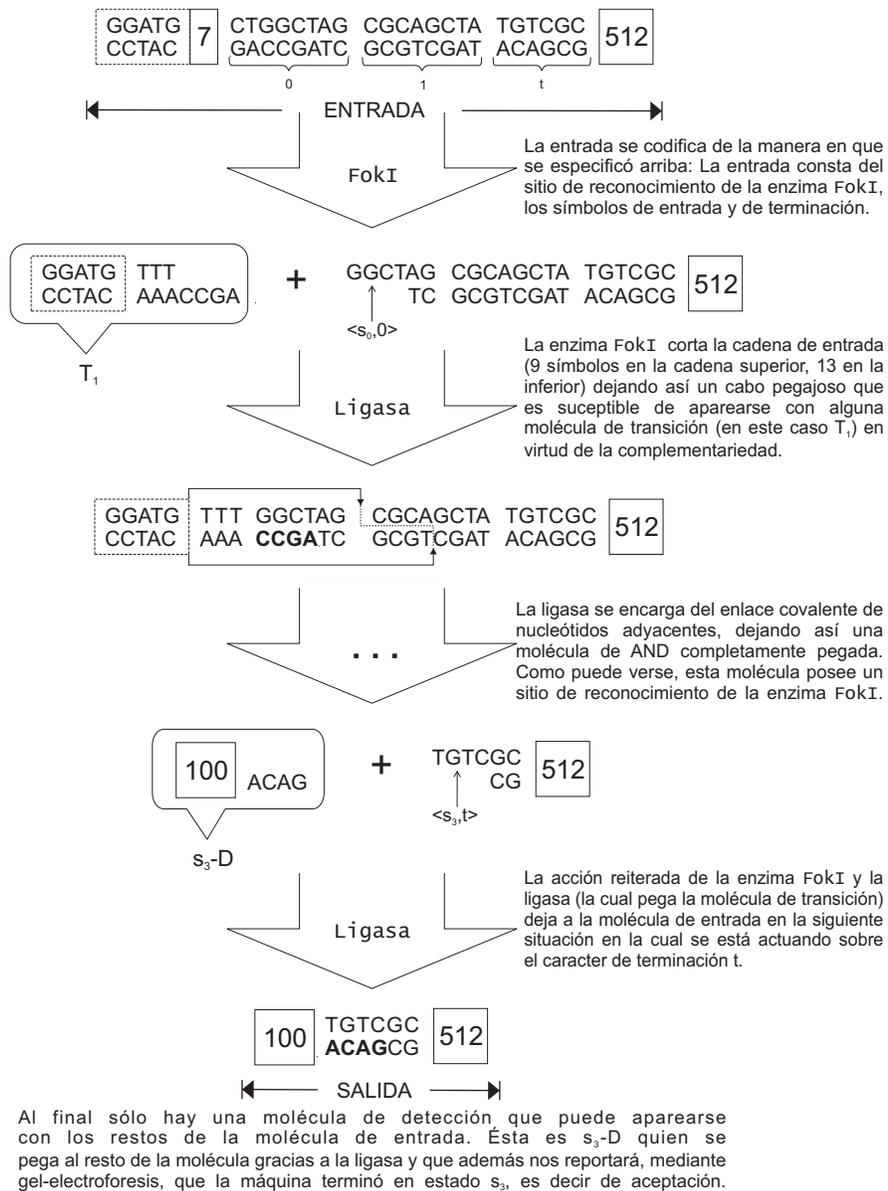


Figura 1.6: Corrida

Capítulo 2

Preliminares

A continuación, revisaremos los conceptos básicos que se necesitan para comprender la exposición del presente trabajo. Repasaremos las operaciones básicas, que se realizan sobre cadenas de ADN, y que se conocen ampliamente además de estudiarse exhaustivamente en la ingeniería genética y las disciplinas relacionadas. Asimismo, haremos una descripción sucinta de un par de aplicaciones de este, relativamente nuevo, modelo de cómputo; las cuales son muy significativas, ya que representan dos grandes tendencias en el estudio de la *computación genética*: una ve en este modelo de cómputo un paradigma para la solución de problemas *difíciles* de la computación electrónica (véase el *Experimento de Adleman* más abajo); mientras que la otra ve en este modelo un paradigma para el combate de diversos padecimientos crónico-degenerativos desde el frente genético (véase el *Doctor ADN* más adelante).

2.1. La Computación Genética

Se sabe desde hace unos años que la continua miniaturización de los dispositivos electrónicos que hacen posible la computación actual tiene consecuencias indeseables; crear *chips* más y más chicos en un área cada vez más pequeña resulta en un incremento incontrolable de la temperatura conforme los dispositivos comienzan a operar. Al parecer los fabricantes de *hardware* están alcanzando el límite físico al querer desarrollar dispositivos más veloces en el menor espacio posible. Ya desde hace algún tiempo los fabricantes de computadoras y de dispositivos de almacenamiento pretenden llegar al *bit por átomo* lo cual se traduce en almacenamiento gigantesco por poco volumen. Sin embargo, la teoría nos dice que para conseguir esto se tiene que pagar el precio *cuántico* que tal pretensión conlleva y así nace la *computación cuántica* cuya finalidad es aprovechar los fenómenos cuánticos que suceden al manipular partículas subatómicas (e.g. electrones o fotones) para hacer cómputo. Otro de los modelos que se ha propuesto como alternativa a la computación electrónica y que también se trata de realizar cómputo a escalas moleculares es la computación genética; esta tesis se refiere a dicho paradigma. La computación genética se refiere a los problemas que se pueden resolver aprovechando las propiedades del material genético: el *Ácido Desoxirribonucleico* (ADN).

Hay dos características del ADN que lo hacen atractivo para la implementación de *software*:

Paralelismo La densidad de información almacenada en cadenas de ADN más la relativa facilidad con la que se construyen copias de la misma abre la posibilidad de llevar a cabo cómputo en paralelo de forma masiva. Por mencionar un par de ejemplos donde esta característica es de gran utilidad: la solución del *Problema del Viajero* (TSP, por sus siglas en inglés) para un número considerable de ciudades (véase Experimento de Adleman más abajo); y el desciframiento de llaves en el criptoanálisis [6].

Complementariedad Una cadena de ADN está compuesta por moléculas llamadas *nucleótidos* los cuales difieren únicamente en un elemento llamado *base*. Existen cuatro bases distintas: *Adenina* (A), *Citocina* (C), *Guanina* (G) y *Tiamina* (T). La famosa doble hélice de ADN aparece al unir dos cadenas de ADN por sus bases. Debido a su composición los nucleótidos con base A se unen siempre con los de base T, y de igual manera, los nucleótidos con base C se unen a los de base G.

El ADN es la molécula central en la computación genética y es crucial en los procesos celulares de los seres vivos. Sus dos principales funciones son la codificación para la creación de nuevas proteínas y la autoreplicación de tal forma que una copia exacta de ella misma es heredada a las células hijas.

La representación de una cadena doble de ADN como dos cadenas simples, una colocada encima de la otra respetando la complementariedad de las bases, es una gran simplificación (la primera que haremos). En la realidad, ambas cadenas están unidas formando una doble hélice; asimismo, en el núcleo de la célula se debe hacer el almacenamiento de ambas cadenas que de ser extendidas en su totalidad, serían más grandes que la propia célula. Entonces, nuestra representación de estas cadenas como una cadena lineal que se extiende horizontalmente nos ayudará en nuestra exposición, aunque se sabe también que las cadenas de ADN de algunas bacterias son circulares, es decir, son cadenas que vienen unidas por los extremos. Este tipo de ADN será empleado en el capítulo 4 y mostraremos la manera de representarlo linealmente.

Ahora, lo siguiente que podemos preguntarnos es: ¿Qué se puede hacer con moléculas de ADN? En el caso de una computadora convencional, los transistores integrados a una pastilla resuelven operaciones aritméticas y lógicas con corrientes eléctricas a partir de operaciones muy sencillas como *mover* un registro de memoria, *sumar* dos registros, *guardar* un resultado y *comparar* el valor de dos registros, por citar algunas. Así, es natural que nos preguntemos por aquello que puede hacerse con cadenas de ADN, en cuyo caso es el estudio de la química orgánica lo que marca el paso. En la siguiente sección expondremos algunas técnicas y descubrimientos para la manipulación de ADN, los cuales son un *regalo* de la bioquímica a este nuevo modelo de cómputo.

2.2. Operaciones con ADN

A continuación describimos brevemente algunas operaciones que pueden llevarse a cabo con ADN. (Una descripción exhaustiva de la forma en que se consiguen las siguientes operaciones está más allá de los objetivos de esta tesis; para ello, se remite al lector interesado al libro [6].)

Medición de una cadena de ADN La longitud de una cadena sencilla de ADN

(es decir, sin aparear) es el número de nucleótidos que componen la cadena. La longitud de una cadena doble de ADN se cuenta por el número de pares de bases que integran la molécula y se mide en *bps* (pares de bases). Para realizar esto se utiliza una técnica llamada *gel electroforesis*, la cual se basa en el hecho de que las moléculas de ADN están cargadas negativamente y que las más pequeñas se desplazan más rápido en un gel.

Separar y unir cadenas de ADN Las cadenas de ADN están unidas por un enlace covalente de manera *horizontal* (cuando los nucleótidos son adyacentes en una cadena sencilla); y por un enlace de hidrógeno cuando se trata de bases complementarias (*verticalmente* en una cadena doble de ADN). Este último es más débil que el primero, y es posible separar dos cadenas complementarias, sin separar los nucleótidos que integran las cadenas sencillas, simplemente aumentando la temperatura. Ahora, si la solución se pone a enfriar, las cadenas separadas son susceptibles de volverse a unir; proceso que debe hacerse lentamente para que las bases complementarias tengan oportunidad de identificarse. Este proceso será denominado *apareamiento* a lo largo de este trabajo.

Extracción de una cadena sencilla de ADN Es posible extraer una cadena sencilla de ADN de entre una multitud de otras moléculas (algo así como extraer la aguja del pajar), lo cual se consigue mediante una técnica de laboratorio, llamada *separación por afinidad* que hace uso de balines microscópicos (aproximadamente de un micrómetro de diámetro) a los cuales se pegan copias de la cadena complementaria que se pretende extraer del total (estas cadenas complementarias usadas así son llamadas moléculas de sondeo). Así, cualquier cadena (simple o doble) que contenga como subcadena la cadena sencilla complementaria a aquella pegada al balín, se adherirá a éste (en virtud de la complementariedad de las bases), permitiendo, a través de un imán, separar estas cadenas de la solución total. Se podría pensar que esta técnica es poco útil pues restringe su uso a cadenas sencillas, pero si se tiene en mente la operación descrita arriba, se puede separar una cadena doble para obtener dos cadenas sencillas y así extraer de la solución sólo aquella que nos interesa del par.

Ahora pasamos a considerar la acción de algunas enzimas sobre una molécula de ADN. Las enzimas son proteínas que catalizan reacciones químicas que ocurren dentro de la célula. Estas son muy específicas tanto así que la mayoría de ellas catalizan solamente una sola reacción química y lo hace muy eficientemente acelerando las reacciones químicas aproximadamente un trillón de veces [6]. Dada la efectividad y utilidad de las enzimas en la operación con moléculas de ADN, su uso se ha extendido ampliamente en la ingeniería genética. A continuación mostramos algunas de ellas y cómo actúan sobre una cadena de ADN.

Extender una cadena de ADN Se consigue usando una clase de enzimas llamada *polimerasas*, las cuales operan agregando nucleótidos a una cadena de ADN dada.

Recortar una cadena de ADN Se logra mediante la acción de un grupo de enzimas denominadas *nucleasas* las cuales son de dos tipos:

1. **Exonucleasas** Enzimas que recortan una cadena de ADN quitando nucleótido por nucleótido de sus extremos.

2. **Endonucleasas** Contrario a lo anterior, recortan la cadena eliminando los enlaces covalentes en el interior de una cadena (y no limitándose a los extremos como las *exonucleasas*). Este tipo de enzimas son muy específicas, pues difieren en el lugar dónde cortan y la forma en la que lo hacen.

Nos limitaremos a la descripción de las *endonucleasas* ya que son el tipo de enzimas que se usarán en la exposición de nuestros resultados; en particular la subclase de *endonucleasas de restricción* o como mejor se les conoce: *enzimas de restricción*.

Existe una endonucleasa llamada *S1*, la cual actúa únicamente sobre cadenas simples de ADN (o donde haya una cadena doble). El corte se lleva a cabo de forma arbitraria en algún punto de la cadena simple y por eso se dice que esta endonucleasa no es de *sitio específico*. Su acción se ejemplifica a continuación. Consideremos la siguiente cadena de ADN

$$N = \frac{N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8}{\widetilde{N}_1 \widetilde{N}_2 \quad \quad \quad \widetilde{N}_6 \widetilde{N}_7 \widetilde{N}_8} \quad (2.1)$$

donde $N_i \in \{A, C, G, T\}$ para $1 \leq i \leq 8$, y \widetilde{N}_i es su base complementaria (i.e. si $N_i = A$, entonces $\widetilde{N}_i = T$, y si $N_i = C$, entonces $\widetilde{N}_i = G$, y viceversa). Así, al someter la cadena N a la acción de la enzima *S1* obtendríamos las dos subcadenas:

$$N' = \frac{N_1 N_2 N_3}{\widetilde{N}_1 \widetilde{N}_2} \quad N'' = \frac{N_4 N_5 N_6 N_7 N_8}{\widetilde{N}_6 \widetilde{N}_7 \widetilde{N}_8} \quad (2.2)$$

Aunque, como mencionamos arriba, el corte se hace arbitrariamente; éste puede llevarse a cabo también en N_4 o N_5 ; por lo que el resultado en (2.2) es sólo uno de cuatro posibles. Debido a tal incertidumbre en el resultado de aplicar esta enzima, su uso es muy restringido en la ingeniería genética (ver [6]). Por suerte, se ha descubierto un tipo de enzimas, las *endonucleasas de restricción*, cuya acción es mucho más específica: éstas cortan solamente cadenas dobles y únicamente en sitios específicos de la cadena. Una endonucleasa de restricción (de aquí en adelante *enzima de restricción*) se une a la cadena de ADN en una zona particular de la misma denominada con justicia *sitio de reconocimiento* (RS, a lo largo de esta tesis) y eliminará enlaces covalentes en la cadena de ADN: algunas veces dentro del sitio de reconocimiento y otras fuera del mismo, pero siempre de manera específica y controlable.

En el ejemplo (2.1) describimos la acción de la enzima *S1*; a continuación mostraremos otras de estas enzimas y su acción; pero antes es útil mencionar que el nombre de ellas surge del nombre del organismo del cual se aisló por primera vez la enzima; así, por ejemplo, la enzima *EcoRI* (descrita abajo) denota a la primera (I) enzima de restricción aislada a partir de la bacteria *Escherichia coli*, cepa serotipo *R*. Más interesante es mencionar que en los organismos vivos, las enzimas de restricción cumplen una tarea muy importante, la de servir de defensa en contra de otros organismos invasores. Es el caso de una bacteria que identifica un virus nocivo en su interior; las enzimas de restricción *digieren* (cortan en pedazos) el ADN del organismo invasor: una amplia gama de enzimas de restricción permite la defensa contra una amplia gama de invasores [6]. Algunas de las enzimas de restricción que se conocen actualmente son:

EcoRI El sitio de reconocimiento de esta enzima es **GAATTC**. La enzima se unirá a este sitio de reconocimiento y cortará dentro del mismo de la siguiente forma:

$$N = \begin{array}{l} GAATTC \\ CTTAAG \end{array} \quad (2.3)$$

Obteniendo las dos subcadenas tras el corte:

$$N' = \begin{array}{l} G \\ CTTAA \end{array} \quad N'' = \begin{array}{l} AATTC \\ G \end{array} \quad (2.4)$$

A diferencia de la enzima *S1* considerada en (2.1), la enzima *EcoRI* cortará únicamente en el sitio descrito arriba y en ningún otro. El sitio de reconocimiento **GAATTC** es un *palíndromo* en el sentido de que al leerse en la cadena superior de izquierda a derecha se obtiene el mismo resultado que leyéndose en la cadena inferior de derecha a izquierda. En el capítulo 4 será útil considerar los cortes en un sentido y en otro, para la cadena superior e inferior, respectivamente. Como la enzima *EcoRI* corta dentro de este sitio, no tiene sentido hablar de un corte de izquierda (en la cadena superior) a derecha y de derecha a izquierda (en la cadena inferior), pues se obtiene el mismo resultado. Sin embargo sí es necesario mencionar que la enzima corta si se encuentra el sitio **CTTAAG** en la cadena inferior, NUNCA si se encuentra en la cadena superior.

XmaI El sitio de reconocimiento de esta enzima es la cadena **CCCGGG**, que también corta dentro de este palíndromo de la siguiente forma:

$$N = \begin{array}{l} CCCGGG \\ GGGCCC \end{array} \quad (2.5)$$

Tras su acción se obtiene las dos cadenas siguientes:

$$N' = \begin{array}{l} C \\ GGGCC \end{array} \quad N'' = \begin{array}{l} CCGGG \\ C \end{array} \quad (2.6)$$

FokI Esta enzima es usada extensivamente en la investigación de la computación genética y el presente trabajo no será la excepción, ya que con ella conseguiremos resolver nuestras aproximaciones a la función N-AND. El sitio de reconocimiento de la enzima es la cadena **GGATG**, pero su manera de cortar difiere notablemente de las que hasta ahora se habían considerado, pues ella corta *fuera* de su sitio de reconocimiento. Su acción se muestra en la siguiente cadena:

$$N = \begin{array}{l} GGATG \quad \underbrace{N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14}} \\ CCTAC \quad \underbrace{\bar{N}_1 \bar{N}_2 \bar{N}_3 \bar{N}_4 \bar{N}_5 \bar{N}_6 \bar{N}_7 \bar{N}_8 \bar{N}_9 \bar{N}_{10} \bar{N}_{11} \bar{N}_{12} \bar{N}_{13} \bar{N}_{14}} \end{array} \quad (2.7)$$

De tal forma que al aplicar la enzima *FokI* obtenemos, por un lado, la cadena:

$$N' = \begin{array}{l} GGATG \quad \underbrace{N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9} \\ CCTAC \quad \underbrace{\bar{N}_1 \bar{N}_2 \bar{N}_3 \bar{N}_4 \bar{N}_5 \bar{N}_6 \bar{N}_7 \bar{N}_8 \bar{N}_9 \bar{N}_{10} \bar{N}_{11} \bar{N}_{12} \bar{N}_{13}} \end{array} \quad (2.8)$$

Y por el otro:

$$N'' = \begin{array}{c} N_{10}\widetilde{N}_{11}N_{12}N_{13}\widetilde{N}_{14} \\ \widetilde{N}_{14} \end{array} \quad (2.9)$$

Es decir, la enzima corta fuera de su sitio de reconocimiento *contando* nueve nucleótidos después de éste sobre la cadena superior y trece sobre la cadena inferior. (De allí que la cadena (2.9) sólo contenga los cuatro nucleótidos que completan la cadena superior de la cadena (2.7)). Como puede verse, el sitio de reconocimiento **GGATG** no es un palíndromo como en las enzimas anteriores y su corte es diferente en la cadena superior y en la inferior, por lo que tiene sentido diferenciar entre un corte que se realiza de izquierda a derecha de otro en sentido opuesto. Si la enzima se encuentra la subcadena **GGATG** sobre la cadena inferior **NO** cortará la cadena doble, sin embargo, si se encuentra su versión especular, la cadena **GTAGG**, **SÍ** realizará el corte pero de derecha a izquierda (véase el capítulo 4).

BseRI Esta enzima actúa de manera similar a *FokI* ya que hace su corte fuera del sitio de reconocimiento, siendo éste la cadena **GAGGAG**. El corte se hace como se ejemplifica a continuación:

$$N = \begin{array}{c} GAGGAG \ N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10}N_{11}N_{12}N_{13} \\ CTCCTC \ \widetilde{N}_1\widetilde{N}_2\widetilde{N}_3\widetilde{N}_4\widetilde{N}_5\widetilde{N}_6\widetilde{N}_7\widetilde{N}_8\widetilde{N}_9\widetilde{N}_{10}\widetilde{N}_{11}\widetilde{N}_{12}\widetilde{N}_{13} \end{array} \quad (2.10)$$

Al aplicar la enzima obtenemos las siguientes cadenas:

$$N' = \begin{array}{c} GAGGAG \ N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10} \\ CTCCTC \ \widetilde{N}_1\widetilde{N}_2\widetilde{N}_3\widetilde{N}_4\widetilde{N}_5\widetilde{N}_6\widetilde{N}_7\widetilde{N}_8 \end{array} \quad (2.11)$$

Y la cadena:

$$N'' = \begin{array}{c} N_{11}N_{12}N_{13} \\ \widetilde{N}_9\widetilde{N}_{10}\widetilde{N}_{11}\widetilde{N}_{12}\widetilde{N}_{13} \end{array} \quad (2.12)$$

SmaI Hasta ahora, se han considerado cortes *escalonados*. La enzima *SmaI* hace un corte *al ras* sobre la cadena de ADN, como se verá a continuación. El sitio de reconocimiento de esta enzima es la cadena **CCCGGG**, el mismo que la enzima *XmaI* pero su acción es la siguiente:

$$N = \begin{array}{c} CCCGGG \\ GGGCCC \end{array} \quad (2.13)$$

Tras su acción se obtiene las dos cadenas siguientes:

$$N' = \begin{array}{c} CCC \\ GGG \end{array} \quad N'' = \begin{array}{c} GGG \\ CCC \end{array} \quad (2.14)$$

Se debe mencionar que si una cadena de ADN contiene varios sitios de reconocimiento, en principio, una enzima de restricción cortará todos ellos.

Ahora, consideremos otras operaciones que se puede hacer sobre cadenas de ADN.

Pegar una cadena de ADN Recordemos que hay dos maneras de unir dos nucleótidos: por medio de un *enlace covalente* cuando se trata de nucleótidos adyacentes, es decir, pertenecen a la misma cadena simple (*horizontalmente*); o por medio de un *enlace de hidrógeno* que se logra sólo cuando los nucleótidos son complementarios (*verticalmente*) y es la manera en que se forman las cadenas dobles de ADN. Las moléculas de ADN se pegan unas con otras a partir de un proceso mediado por enzimas llamadas *ligasas* (o *ligasa* en el resto de este trabajo). Para describir su funcionamiento, consideremos la acción de una enzima de restricción como las descritas arriba. Se puede ver en los ejemplos (2.2), (2.4), (2.6), etc. que la acción de la enzima produce dos cadenas con nucleótidos sin aparearse, los cuales son susceptibles de volver a unirse por medio de un enlace de hidrógeno si se les acerca lo suficiente. Estos nucleótidos son conocidos en la literatura como *cabos pegajosos* dada la complementariedad de cada uno. Ahora, no basta con que estos cabos se *peguen* para que dos cadenas sencillas de ADN formen una cadena doble, pues hace falta que los nucleótidos adyacentes se unan mediante un enlace covalente. Esto se consigue gracias a la acción de la *Ligasa*, cuya acción se ve facilitada por la unión de los cabos pegajosos. Así, el enlace de dos cadenas con cortes escalonados se consigue más fácilmente que el enlace de dos cadenas al ras (i.e. sin estos cabos), ya que la *Ligasa* no es capaz de mantener demasiado cerca ambas cadenas para pegarse.

Copiar una cadena de ADN Con el desarrollo de una técnica llamada *reacción en cadena de polimerasa* (PCR, por sus siglas en inglés) se resuelve uno de los mayores problemas de la ingeniería genética; a saber, el de amplificar una cadena de ADN. Por medio de la PCR puede generarse millones de copias de una molécula de ADN en relativamente poco tiempo: el número de copias se duplica a cada paso.

Leer una cadena de ADN Leer el contenido de una cadena de ADN es un proceso complicado conforme la cadena crece de tamaño (ver arriba: medición de una cadena de ADN). Téngase en mente que el *Proyecto del Genoma Humano* tiene como objetivo determinar toda la secuencia de nucleótidos existentes en el genoma, el cual tiene una longitud aproximada de 3×10^9 bps [6]. Describir a detalle cómo se logra la secuenciación de una cadena de ADN (es decir, conocer cada uno de los nucleótidos que componen una cadena) está fuera de los objetivos de esta tesis, sólo mencionaremos que esto es posible a partir de la amplificación de una cadena sencilla que ha sido preparada en laboratorio y que sirve de plantilla, más el uso de nucleótidos modificados *in vitro*.

Con estas operaciones en mente, podemos pasar a describir el experimento que motivó el estudio de esta clase de cómputo: *El Experimento de Adleman*.

2.3. El Experimento de Adleman

En 1994 el científico estadounidense Leonard Max Adleman, conocido anteriormente como uno de los coautores del sistema de cifrado RSA usado ampliamente en aplicaciones de seguridad y firmas digitales, escribió un artículo titulado *Computación molecular de soluciones a problemas combinatorios* donde se describe el

uso del material genético (ADN) como un posible sistema de cómputo. En dicho documento se resuelve una instancia de siete nodos del problema de la trayectoria hamiltoniana para una gráfica dada.

El **Problema de la trayectoria Hamiltoniana** es el de encontrar, en una gráfica dada, una trayectoria que pase por todos y cada uno de los vertices una sola vez. En su aplicación, el problema puede ejemplificarse en la siguiente situación: dado un conjunto de ciudades, y los caminos que las comunican unas con otras, decir si existe o no un camino que parta de una ciudad en particular y termine en otra, pasando por cada una de las ciudades restantes una sola vez. Comúnmente se dice que no existe un algoritmo eficiente que resuelva este problema, tanto así que se ha demostrado que hay ciertas gráficas de no más de cien vértices cuya solución al problema tomaría algunos cientos de años a las computadoras más poderosas de la actualidad [7].

El experimento de Adleman consiste en asignar, a cada una de las ciudades de la gráfica, una cadena simple de ADN de ocho nucleótidos de longitud, llamados *oligos*, de tal forma que a través de esta cadena se puede identificar un camino que va de una ciudad a otra de la siguiente manera: los cuatro últimos nucleótidos de una cadena (que representa una ciudad) denotan la ciudad origen concatenados a los cuatro primeros nucleótidos de la ciudad destino [7]. En otras palabras, el nombre del camino que comunica dos ciudades está dado por los cuatro últimos nucleótidos que representan la ciudad origen más los cuatro primeros nucleótidos que representan la ciudad destino. Por ejemplo, si la ciudad A está representada por la cadena *ACTTGCAG* y la ciudad B por la cadena *TCGGACTG*, entonces la cadena doble que representa el camino que va de la ciudad A a la ciudad B es:

ACTTGCAGTCGGACTG
CGTCAGCC

Adleman plantea el siguiente algoritmo para la solución de este problema que después adaptará molecularmente:

1. Generar un conjunto de trayectorias de la gráfica aleatoriamente.
2. Para cada trayectoria de este conjunto hacer:
 - a) Revisar si la trayectoria comienza y termina en las ciudades esperadas, de lo contrario desechar esa trayectoria.
 - b) Revisar que la trayectoria pase por el número de ciudades de la gráfica; no más no menos, de lo contrario desechar esa trayectoria.
 - c) Para cada uno de las ciudades, revisar que esa trayectoria pase por ella, de lo contrario desechar la trayectoria.
3. Si al final del proceso el conjunto no es vacío, entonces reportar que existe una trayectoria Hamiltoniana para esa gráfica.

Ahora, la versión molecular del algoritmo anterior explota las propiedades del ADN que hemos venido describiendo hasta este momento. A continuación se describe cómo Adleman implementa el algoritmo en moléculas de ADN:

1. Se sintetizan las cadenas complementarias de cada una de las cadenas que representan las ciudades más las cadenas que denotan el camino que hay de una ciudad a otra de acuerdo a la gráfica (o al mapa de las ciudades).

(Según Adleman, es posible escribir una cadena de ADN en papel, enviarla a un laboratorio dedicado a la síntesis comercial y en unos cuantos días obtener un tubo de ensayo con aproximadamente 10^{18} moléculas de ADN las cuales todas, o una gran mayoría tienen la secuencia deseada [7].) El cómputo comienza al mezclar agua, las moléculas que representan los caminos de ciudad a ciudad, *Ligasa*, sal y otros ingredientes para simular el interior de una célula; obteniendo una mezcla no mayor a un quinto del tamaño de una cuchara de té.

2. Para la solución obtenida mezclando las cadenas complementarias de cada ciudad y las cadenas que denotan el camino de una ciudad a otra, hacer:
 - a) Amplificar las cadenas, mediante PCR (ver arriba), que contienen los últimos cuatro nucleótidos de la ciudad inicial en un extremo y los primeros cuatro de la ciudad final en el otro.
 - b) Usando la técnica de *gel electroforesis* (ver arriba) se identifican las cadenas que tienen la longitud correcta ($8 \times (\text{ciudades} - 1)$) medidas en número de nucleótidos.
 - c) Ya que en el paso anterior se descartó todo aquello que no comenzara con la ciudad inicial y terminara en la ciudad final, restaría saber si se recorrieron las ciudades intermedias. Para saberlo se usa la técnica de *separación por afinidad* (ver arriba), pegando los complementos de las cadenas que representan las ciudades intermedias a los balines. Este paso es el más laborioso [7], ya que para cada una de las ciudades intermedias se debe repetir el siguiente proceso: se *construye* un balón con la cadena complementaria de una ciudad intermedia; se introduce el balón a la solución total para que la molécula que tiene como subcadena aquella que representa la ciudad intermedia se adhiera al balón en virtud de la complementariedad de sus bases; mediante la acción de un imán se separan estas cadenas de la solución total, la cual consiste ahora de moléculas que NO contienen como subcadena la ciudad intermedia y por tanto puede desecharse. Las moléculas que SÍ se pegaron al balón son separadas de éste usando un solvente y calentando la solución (para separar las cadenas dobles que se llegaron a formar); este proceso se repite para las ciudades intermedias restantes.

Obsérvese que lo anterior debe repetirse una vez por cada vértice de la gráfica.

3. Cuando concluye la revisión de las ciudades intermedias se puede confiar en que las cadenas que aún permanecen en el tubo de ensayo (en el *Experimento de Adleman* sí quedaron moléculas en el tubo) corresponden no sólo a cadenas que empiezan en la ciudad inicial y terminan en la ciudad final, ni sólo moléculas que pasaron exactamente por el número de ciudades de la gráfica, sino que además se trata de cadenas que corresponden a trayectorias que pasaron por cada una de las ciudades intermedias así que si algo resta en el tubo de ensayo se trata nada más y nada menos que de trayectorias Hamiltonianas de la gráfica dada. (El caso en que sean ciudades intermedias unidas por caminos inexistentes, es decir, por aristas que no aparecen en la gráfica, queda descartado desde el inicio pues debe recordarse que la codificación de las cadenas se realizó en base a los caminos de una ciudad a otra.)

Como puede verse, la magia del *Experimento de Adleman* radica en la posibilidad de probar miles de trayectorias a la vez (*paralelismo*) pues como el mismo Adleman dice: *como se inició el proceso con un gran número de moléculas* (aproximadamente 10^4 moléculas por cadena-camino y por cadena-complementaria-ciudad) *y el problema contiene solamente un puñado de ciudades, hay una certeza virtual de que al menos una de las moléculas formadas codifique una trayectoria Hamiltoniana* [7]. Este proceso es equivalente a tener miles de personas o computadoras intentando una trayectoria a la vez. Y así, como se puede ver, las propiedades del material genético ayudan a la realización del proceso, más que realizar el cómputo mismo.

Un resultado interesante sería el que estableciera una relación entre el número de ciudades y la cantidad mínima de moléculas de ADN que se deben considerar para esperar una solución al problema, ya que la ausencia de una cadena de ADN al final de proceso no se debe interpretar *directamente* como la imposibilidad de una trayectoria Hamiltoniana para una gráfica dada (podría ocurrir que NO se administró el número adecuado de moléculas al iniciar el proceso). Pero por el momento, el *Experimento de Adleman* sirve para motivarnos en el estudio de la computación genética no sólo como medio alternativo de cómputo, es decir, por su capacidad de resolver problemas de gran complejidad con relativa facilidad, sino adicionalmente para ampliar el entendimiento sobre la manera en que la Naturaleza realiza sus procesos, pues recordemos que la sola manipulación del ADN resulta en la extraordinaria complejidad fenotípica de los seres vivos. Un ejemplo de esto es el *Autómata de Benenson-Shapiro* que fue diseñado para el control de la expresión genética celular *in vitro* [3].

2.4. El Autómata de Benenson-Shapiro: el Doctor ADN

Ehud Shapiro y Yaakov Benenson nos ofrecen otro ejemplo de la aplicación de la computación genética que quizá va más de acuerdo a la naturaleza de sus medios de cómputo. La aplicación que ellos sugieren motiva e inspira el estudio de esta clase de cómputo, ya que se trata de un programa, implementado en ADN, para el diagnóstico de algunas enfermedades degenerativas [3, 4]. Hasta ahora el equipo de estos científicos ha logrado detectar correctamente *in vitro* la presencia de moléculas relacionadas con enfermedades cancerígenas, en la mayoría de las veces en que fue probada, permitiendo además el suministro controlado de fármacos para su tratamiento [3]. En resumen, su trabajo se llevó a cabo como se describe a continuación.

Estos investigadores primero vislumbraron la creación de un *autómata* hecho a partir de moléculas de ADN [4]; usando como *hardware* las enzimas *FokI* y *Ligasa* más *ATP* que sirve como fuente de energía de la máquina, mientras que para el *software* diseñaron un conjunto de moléculas de ADN (cadenas dobles) de poco tamaño que codifican las reglas de transición del autómata (por ello, se les denomina moléculas de transición). Por último, la entrada de este autómata consiste de las cadenas que se pueden formar a partir de dos caracteres, donde estos caracteres han sido codificados molecularmente (más un caracter de terminación) [2]. Debe decirse en este punto que gran parte del trabajo desarrollado en el capítulo 3 está inspirado en los resultados de esos autores. La *máquina* de ADN funciona *digiriendo* la cadena de entrada; es decir, el *hardware* lleva a cabo su función cor-

tando la cadena exponiendo un cabo que es susceptible de pegarse mediante *Ligasa* a alguna molécula de transición y así desechar el símbolo que acaba de ser *leído* y pasar al siguiente.

En [2] se presenta un conjunto de autómatas en su versión abstracta, y se muestra la implementación de uno de ellos con ADN (programado para responder si la entrada contiene un número par de uno de los símbolos del alfabeto). Esta máquina corre de manera autónoma en el sentido de que al mezclar la entrada, el *hardware* y el *software* en una solución apropiada, dentro de un tubo de ensayo, el cómputo (la serie de cortes y pegados) se realiza sin intervención humana, paso por paso, hasta concluir el proceso; es decir, hasta que no se encuentren más sitios de reconocimiento en la cadena de entrada. Así, este equipo de investigadores fue capaz de simular usando ADN, una máquina capaz de responder si su entrada posee una propiedad en particular. Lo siguiente era aplicar esa misma idea para diagnosticar una enfermedad que se manifiesta en el interior de la célula. La motivación es la siguiente: los organismos vivos llevan a cabo procesos físicos que obedecen ciertas reglas codificadas en su material genético; así, podríamos comparar la acción de una *Máquina de Turing* (definida más adelante) con la de algunos sistemas dentro de la célula que manipulan y procesan cadenas de ADN (y ARN: *ácido ribonucleico*), descubriendo que ambas operan paso por paso sobre cadenas de un alfabeto en particular, modificando su contenido y de acuerdo a reglas específicas [4].

No obstante, se sabe que el poder computacional (en términos de rendimiento) de una computadora electrónica sobrepasa enormemente al de cualquier que pudiera imaginarse. Dicho sea de paso, la velocidad de un organelo como el *ribosoma* es de cientos de operaciones por segundo, mientras que algunas computadoras ejecutan billones de operaciones por segundo actualmente. Nuestra perspectiva cambia cuando consideramos la posibilidad de realizar operaciones de forma paralela (como en la sección anterior), o al considerar que estas nuevas máquinas *hablan* el lenguaje nativo de las células [4].

El autómata que desarrollaron Benenson, Shapiro y colegas, tras probar suerte con autómatas abstractos, tendría la capacidad de responder a partir de los cambios en el medio celular, percibiendo la presencia de moléculas de ARN mensajero que, debido a su abundancia o escasez, indican la presencia de anomalías en la función celular. Dicha máquina cuenta únicamente con dos estados, *SÍ* y *NO*, siendo el primero el estado inicial el cual no se modifica hasta que una de las condiciones del ambiente no está presente. Este cambio se realiza a través de la acción de las moléculas de transición; éstas se encuentran *inactivadas* por una molécula que sólo puede ser removida si en el medio celular se encuentran moléculas de mARN (ARN mensajero) que indican la sobreexpresión de un gen. La máquina está diseñada de tal forma que al responder a la presencia de ciertas moléculas de mARN (vía las moléculas de transición) permiten la liberación de una molécula de ADN, que sirve como medicamento, o la liberación de su antagonista, inhibiendo la acción del fármaco y regulando la acción total de las máquinas que paralelamente realizan este análisis.

El hecho que motiva este paradigma de la lucha contra el cáncer, es la presencia de moléculas de mARN (y últimamente se ha venido estudiando acerca de una nueva clase de moléculas llamadas *microARN* [4]) que se traduce en niveles anormales de ciertas proteínas que caracterizan varios cánceres [3]. Sin embargo, debe mencionarse que esta investigación apenas está dando sus primeros pasos, pues como se mencionó al inicio de la sección, los resultados de dicho estudio apenas se han dado

in vitro. La implementación de este tipo de máquinas en una célula viva requiere, entre otras cosas, la creación de mecanismos de control que prevengan que la acción de la máquina afecte el funcionamiento ordinario de la célula; no obstante, parece ser un paradigma muy promisorio.

2.5. DNA-N-AND

DNA-N-AND no sólo es un palíndromo, es el tema central del presente trabajo que tiene por objetivo la implementación en ADN de la compuerta lógica N-AND, y como se sabe, una compuerta lógica es el dispositivo de cómputo más elemental en las computadoras convencionales. Pero en lo sucesivo nos olvidaremos de los aspectos químicos y físicos que subyace a la Computación Genética y nos quedaremos únicamente con la esencia de las operaciones descritas arriba más las propiedades del material genético consideradas en abstracto.

En concreto, nos interesa conocer los problemas que pueden ser resueltos al ser modelados por una cadena cuyos caracteres sólo pueden tomar uno de cuatro valores posibles (A, C, G o T), con la propiedad de que cada uno de esos cuatro valores es complementario a uno de los tres restantes, y donde esa *complementariedad* se traduce en una capacidad de relacionarse con una segunda cadena (obviamente, de las mismas características que la primera) creando el concepto de cadena doble cuando estas cadenas se consideran unidas por sus caracteres complementarios; y permitiendo solamente la unión de dos cadenas de esta naturaleza y no más. Además de esto se introduce una condición adicional a este modelo axiomático: si en algún lugar de la cadena doble (suponiendo el apareamiento de la cadena inicial con su cadena complementaria) se encuentra una subcadena con una composición específica, la cadena doble es susceptible de cortarse determinado número de caracteres después de la subcadena particular y dado que el número de caracteres que corta en la cadena es distinto de aquél en su cadena complementaria se abre la posibilidad de que la cadena doble recién cortada sea concatenada a otra en virtud de la complementariedad del *cabo* que resulta tras el corte.

Esto es, en pocas palabras, lo que la matemática ve en la computación genética. Y aquí mostraremos cómo una arreglo particular de estas cadenas, al ser apareado con otras del mismo tipo, logra imitar perfectamente el comportamiento de una máquina matemática que a su vez, y en abstracto, *logra imitar perfectamente* el comportamiento de la función booleana N-AND, en las dos versiones presentadas al inicio del capítulo pasado.

En el capítulo 3 se presentará un autómata similar al desarrollado por Benenson y Shapiro, que resuelve la función NAND en su *1a Aproximación* (ver arriba). Para esto se usará un autómata con cinco estados y un alfabeto de dos símbolos (0 y 1) adoptando el concepto de *byte* formado a partir de nucleótidos (de símbolos del conjunto $\Sigma = \{A, C, G, T\}$); se describirá ampliamente la noción de *molécula de transición* y de *molécula de detección* que no son otra cosa que cadenas formadas del conjunto Σ y que realizan el cómputo. Para explicar y probar que estas moléculas en verdad simulan la función N-AND, adoptaremos una notación preliminar; una que aún menciona explícitamente la composición de las moléculas de transición, detección y de entrada, para eventualmente llegar a una notación que sólo hace mención implícitamente de los sitios de reconocimiento de las enzimas de restricción y de las cadenas de entrada (capítulo 4). De tal forma que en el capítulo siguiente se describirá el funcionamiento de una máquina de ADN que simula perfectamente

la función N-AND en la *1a Aproximación* para una entrada pequeña ($n = 2$) a manera de introducción a la computación genética; más adelante, se resolverá la generalización ($n \geq 2$) usando un autómata un poco más complejo y por tanto, otras moléculas de entrada, transición y detección.

Ya para el capítulo 4 nos debe ser claro la forma en que se lleva a cabo el cómputo con ADN, cuáles sus limitantes y cuáles sus ventajas, y dónde reside el problema de diseñar un algoritmo para una máquina molecular; en dicho capítulo se resolverá la *2a Aproximación* de la función NAND, la cual, como se mencionó al principio, es más parecida a la noción de procesamiento de una computadora convencional. Para conseguir esto presentaremos una Máquina de Turing (definida más adelante) que, en abstracto, simula perfectamente la función; y más adelante se presentará y probará la implementación de esta máquina en ADN. Para este entonces ya habremos madurado una notación que prescinde lo más posible del concepto de base y de nucleótido.

Al final, cuando ambas *Aproximaciones* hayan sido resueltas molecularmente, se expondrán las conclusiones a las que se llegaron, así como la discusión que estos resultados puedan generar.

Capítulo 3

1a Aproximación

Una vez que hemos revisado las operaciones básicas que se realizan sobre cadenas de material genético, estamos en condiciones de entender la forma en que se resolverá, mediante ADN, la primera versión de la función lógica N-AND.

Llamemos de ahora en adelante a la primera aproximación de la función N-AND, $NAND_1$ y recordemos que se trata de una función

$$f : \{0, 1\}^n \rightarrow \{0, 1\} \quad n \geq 2 \quad (3.1)$$

Recordemos que esta función lee, de izquierda a derecha, una cadena de 1s y 0s computando la función N-AND de dos en dos y escribiendo el resultado de esta función sobre la misma cadena (ver Introducción); obteniendo, en cada paso, una cadena más corta que la original. Para hacer más clara la exposición, y de paso irnos habituando al mecanismo de este modelo de cómputo, primero nos enfocaremos en resolver el caso más simple de esta función lógica, a saber: el caso en que solamente hay dos *bits* de entrada; para, posteriormente, entrar de lleno en la generalización de la primera *Aproximación*.

Enfocándonos únicamente en la *1a Aproximación* y el caso más simple, cuando $n = 2$, la función es básicamente la tabla de verdad de la función N-AND como se describe en la tabla (3.1). A continuación mostraremos cómo se simula la acción de dicha función mediante moléculas de ADN. En resumen, se simulará la acción de un *Autómata Determinista Finito*, cuyo programa (*software*) realiza la función N-AND sobre una cadena binaria codificada en moléculas de ADN. Para ello, será necesaria no sólo la codificación de la entrada en cadenas de ADN, sino también de las transiciones y estados de la máquina.

Cuadro 3.1: Tabla de Verdad de la función N-AND.

p	q	p NAND q
0	0	1
0	1	1
1	0	1
1	1	0

3.1. $NAND_1$ con $n = 2$

Definición 3.1.1. Un *Autómata Determinista Finito* (DFA) es una quintupla (S, Σ, T, s_0, A) que consiste en:

- Un conjunto finito de estados (S).
- Un conjunto finito llamado alfabeto (Σ). Los elementos de Σ se conocen como símbolos.
- Una función de transición ($T : S \times \Sigma \rightarrow S$).
- Un estado inicial ($s_0 \in S$).
- Un conjunto finito de estados de aceptación ($A \subset S$).

Definición 3.1.2. Sea M un DFA tal que $M = (S, \Sigma, T, s_0, A)$ y σ una cadena formada de elementos del alfabeto Σ , esto es, $\sigma = a_0a_1a_2 \dots a_n$ con $a_i \in \Sigma \forall i$. Se dice que M acepta la cadena σ si existe una secuencia de estados $s_0s_1s_2 \dots s_n$ (con $s_i \in S \forall i$) que cumple las siguientes condiciones:

- $s_i = T(s_{i-1}, a_{i-1})$ con $i = 1, \dots, n$.
- $s_n \in A$.

Si no cumple alguna de estas condiciones, se dice que M rechaza la entrada σ .

La acción de la función $NAND_1$ (con $n = 2$) sobre los caracteres de entrada puede simularse a través de un DFA en el cual un estado de aceptación se asocia con el valor 1 y cualquier otro con el valor 0. Nuestro autómata $M = (S, \Sigma, T, s_0, A)$ está definido de la siguiente forma:

- $S = \{S_0, S_1, S_2, S_3, S_4\}$.
- $\Sigma = \{0, 1\}$.
- La función de transición T se define como:
 1. $T(S_0, 0) = S_1$
 2. $T(S_0, 1) = S_2$
 3. $T(S_1, 0) = S_3$
 4. $T(S_1, 1) = S_3$
 5. $T(S_2, 0) = S_3$
 6. $T(S_2, 1) = S_4$
- El estado S_0 es el estado inicial.
- $A = \{S_3\}$.

Notación 3.1.1. En lo sucesivo representaremos la transición $T(s_i, a) = s_{i+1}$ como $T_k : s_i \xrightarrow{a} s_{i+1}$

El diagrama de estados de este DFA se presenta en la figura (3.1), donde S_3 representa el estado de aceptación que será interpretado como el valor 1 de salida.

En la figura anterior observamos las siguientes transiciones:

$$T_1 : S_0 \xrightarrow{0} S_1 \quad (3.2)$$

$$T_2 : S_0 \xrightarrow{1} S_2 \quad (3.3)$$

$$T_3 : S_1 \xrightarrow{0} S_3 \quad (3.4)$$

$$T_4 : S_1 \xrightarrow{1} S_3 \quad (3.5)$$

$$T_5 : S_2 \xrightarrow{0} S_3 \quad (3.6)$$

$$T_6 : S_2 \xrightarrow{1} S_4 \quad (3.7)$$

En un DFA el *software* consiste de las reglas de transición las cuales describen el estado en el que se encontrará la máquina (el autómata) tras la lectura de un símbolo de entrada a partir del estado previo.

El diseño del DFA molecular, que se propone en este trabajo, incorpora ideas del autómata de Benenson y Shapiro [2] en el cual el *hardware* consiste de una mezcla de nucleasa de restricción (enzima de restricción) clase IIS, llamada *FokI*. Debe recordarse que en este texto sólo se considera la abstracción del modelo de cómputo con ADN, ya que su implementación física requiere de sustancias y métodos adicionales como es el caso de las moléculas de *ATP*, las cuales suministran energía al sistema para realizar el proceso, y de moléculas de *Ligasa*, para lograr que las moléculas que resultan de la acción de la enzima de restricción, puedan *pegarse* con otras. De hecho, Benenson y Shapiro han llegado a descubrir que es posible prescindir de la *Ligasa* (y por tanto, del *ATP*, ya que estas enzimas son las únicas que requieren de *ATP* para su funcionamiento); así, estos investigadores lograron construir un DFA molecular que no sólo no requería de una fuente de energía, sino que asimismo permitía reusar las moléculas de transición [4]. Con esto en mente, no es tan atrevido que prescindamos de la *Ligasa* en nuestra exposición.

Recordemos cómo actúa la enzima de restricción *FokI*. Esta enzima reconoce la cadena doble *GGATG* (en este trabajo se hace alusión a una cadena doble a partir de la parte *superior* de la cadena, sin necesidad de mencionar su cadena

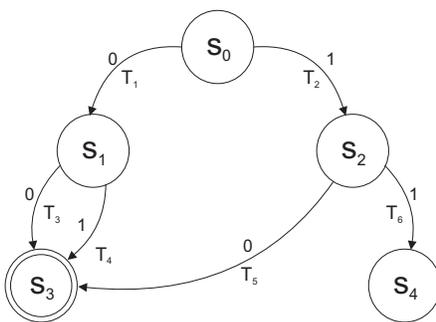


Figura 3.1: Autómata Determinista finito

complementaria). Una vez que la enzima reconoce dicha cadena, *cuenta* 9 bases a la derecha de la cadena superior y 13 bases a la derecha de la cadena inferior. De tal manera que después de cortar se tienen 2 cadenas de ADN cada una con un cabo pegajoso susceptible de volverse a enlazar si las condiciones lo permiten.

El software de la máquina, es decir el programa que será realizado por el DFA, consiste de 6 pares de moléculas base (*bps* o pares de bases) de ADN, las cuales codifican las 6 posibles transiciones descritas en el diagrama del DFA (ver arriba).

Como se verá más adelante, la esencia del autómata, la manera en que se resuelve la función $NAND_1$, reside en estas moléculas de transición y es en el diseño de éstas donde se encuentra el verdadero problema del desarrollo de autómatas implementados en ADN.

Las moléculas de transición son las siguientes (más adelante se describirá la manera de deducirlas y cómo ellas resuelven la función $NAND_1$ que buscamos):

$$T_1 : \begin{array}{|c|c|c|} \hline \text{RS} & 3 & \frac{\text{---}}{\text{CCGA}} \\ \hline \end{array} \quad (3.8)$$

$$T_2 : \begin{array}{|c|c|c|} \hline \text{RS} & 2 & \frac{\text{---}}{\text{GTCG}} \\ \hline \end{array} \quad (3.9)$$

$$T_3 : \begin{array}{|c|c|c|} \hline \text{RS} & 1 & \frac{\text{---}}{\text{GACC}} \\ \hline \end{array} \quad (3.10)$$

$$T_4 : \begin{array}{|c|c|c|} \hline \text{RS} & 1 & \frac{\text{---}}{\text{GCGT}} \\ \hline \end{array} \quad (3.11)$$

$$T_5 : \begin{array}{|c|c|c|} \hline \text{RS} & 2 & \frac{\text{---}}{\text{ACCG}} \\ \hline \end{array} \quad (3.12)$$

$$T_6 : \begin{array}{|c|c|c|} \hline \text{RS} & 0 & \frac{\text{---}}{\text{CGTC}} \\ \hline \end{array} \quad (3.13)$$

Donde, para simplificar la exposición de estas moléculas, se adoptaron las siguientes convenciones:

RS

Denota el sitio de reconocimiento de la enzima de restricción *FokI* (RS, por sus siglas en inglés).

n

Quiere significar que hay n pares-base en esa posición, denominados *nucleótidos espaciadores* [4]. (No estamos interesados en *qué* pares son, sino en *cuántos* son.)

$N_1N_2N_3N_4$

Denota el *cabo pegajoso* $N_1N_2N_3N_4$, es decir, una cadena simple con bases $N_1N_2N_3N_4$ susceptible de acoplarse a su bases complementarias (ver Introducción).

De tal forma que la molécula de transición

$$T_1 : \begin{array}{|c|c|c|} \hline \text{RS} & 3 & \frac{\text{---}}{\text{CCGA}} \\ \hline \end{array}$$

es la forma simplificada (abreviada) de la molécula:

$$T_1 : \begin{array}{|c|c|c|} \hline \text{GGATG} & \text{TGC} & \text{—} \\ \hline \text{CCTAC} & \text{ACG} & \text{CCGA} \\ \hline \end{array}$$

Donde se puede ver que la primera parte de la cadena (la primer columna) es el sitio de reconocimiento de la enzima de restricción *FokI*, la segunda parte son 3 pares arbitrarios y la última parte de la molécula son 4 bases simples, i.e. sin su complemento (que para el ejemplo anterior sería la cadena *GGCT*). Pronto se verá que a pesar de que se puede escoger cualquier combinación de pares para ocupar la segunda parte de la molécula, se debe cuidar que su elección no choque con el funcionamiento de la máquina; en otras palabras, que lo que se ponga en esa posición no afecte el funcionamiento de la máquina de ADN, como podría suceder si se colocasen moléculas que son reconocidas por la enzima de restricción.

Sea $\Gamma = \{T_1, T_2, T_3, T_4, T_5, T_6\}$ el conjunto de las moléculas de transición como se describieron arriba.

En el capítulo pasado recordamos que las funciones booleanas actúan sobre cadenas de 1s y 0s, que podemos llamar *bytes*. Ahora bien, para implementar esto en nuestra máquina de ADN, debemos elegir una manera de representar los valores de 1 y 0 mediante pares-base de ADN lo cual nos lleva a determinar la *longitud de palabra* de la máquina (tomando prestado *otro* término de las ciencias de la computación). Para esta primera aproximación, la resolución de $NAND_1$, hemos escogido una longitud de palabra de 8 pares para los bytes que representan los símbolos 1 y 0, y una longitud de 6 pares para el carácter que denota el fin de la entrada. Los bytes para representar estos símbolos son los descritos en la tabla (3.2).

Cuadro 3.2: Cadenas de ADN que representan los símbolos 1, 0 y t .

1	0	t
CGCAGCTA	CTGGCTAG	TGTCGC
GCGTCGAT	GACCGATC	ACAGCG

Además de lo anterior, se quiere hacer una relación entre los cabos pegajosos y los estados del DFA; es decir, que cuando uno de los bytes de entrada (ya sea 0, 1 o el carácter de terminación t) sea cortado por la enzima de restricción *FokI*, el cabo pegajoso reporte el estado actual de la máquina. Con esto conseguiremos que el byte dé información del estado actual de la máquina y del carácter que está siendo leído actualmente. De tal manera que para los símbolos 1 y 0 esperamos leer los estados S_0 , S_1 y S_2 , dejando los estados finales, S_3 y S_4 para ser codificados en el carácter de terminación t .

La manera en que vamos a codificar estos estados en los bytes que representan los símbolos 1 y 0 es la siguiente. Para el byte

$$N = \frac{N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8}{\widetilde{N}_1 \widetilde{N}_2 \widetilde{N}_3 \widetilde{N}_4 \widetilde{N}_5 \widetilde{N}_6 \widetilde{N}_7 \widetilde{N}_8}$$

donde \widetilde{N}_i denota el complemento de la base N_i y donde el byte anterior puede representar o el símbolo 0 ó el símbolo 1; tenemos que:

1. Cuando el cabo pegajoso representa el estado S_1 la cadena se ve:

$$\begin{array}{c} N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 \\ \widetilde{N_5 N_6 N_7 N_8} \end{array} \quad (3.14)$$

(El cabo pegajoso es $N_1 N_2 N_3 N_4$ ante la entrada del símbolo N .)

2. Cuando el cabo pegajoso representa el estado S_2 la cadena se ve:

$$\begin{array}{c} N_2 N_3 N_4 N_5 N_6 N_7 N_8 \\ \widetilde{N_6 N_7 N_8} \end{array} \quad (3.15)$$

(El cabo pegajoso es $N_2 N_3 N_4 N_5$ ante la entrada del símbolo N .)

3. Cuando el cabo pegajoso representa el estado S_0 la cadena se ve:

$$\begin{array}{c} N_3 N_4 N_5 N_6 N_7 N_8 \\ \widetilde{N_7 N_8} \end{array} \quad (3.16)$$

(El cabo pegajoso es $N_3 N_4 N_5 N_6$ ante la entrada del símbolo N .)

Ahora, para el símbolo de terminación t

$$\begin{array}{c} TGTCGC \\ ACAGCG \end{array}$$

se dan los siguientes casos:

1. Cuando el cabo pegajoso representa el estado S_3 la cadena se ve:

$$\begin{array}{c} TGTCGC \\ CG \end{array} \quad (3.17)$$

(El cabo pegajoso es $TGTC$ ante la entrada del símbolo t .)

2. Cuando el cabo pegajoso representa el estado S_4 la cadena se ve:

$$\begin{array}{c} TCGC \\ - \end{array} \quad (3.18)$$

(El cabo pegajoso es la cadena $TCGC$ ante la entrada del símbolo t .)

Hablando acerca del símbolo de terminación t , el byte que lo representa sirve para indicarnos el resultado final del proceso y que en la aplicación física (en el laboratorio) de esta máquina, puede ser leído a través del proceso que se conoce como *secuenciación*, o incluyendo terminadores adecuados (de diferentes tamaños) y usando *gel electroforesis*, como se verá más adelante.

En suma, la codificación de estos tres símbolos viene descrita en la figura (3.2). Los bytes que representan los valores de 0, 1 y t , descritos en la tabla (3.2) constituyen el conjunto Σ y que corresponde al *alfabeto* de la máquina.

Símbolo	0	1	t
Versión molecular con codificación de <estado, símbolo> en el cabo pegajoso.	CTGGCTAG 	CGCAGCTA 	<S3,t>

Figura 3.2: Símbolos y su Codificación de los Estados

Ahora definamos el conjunto Δ de *moléculas de detección*. Este conjunto contiene las siguientes moléculas:

$$D_1 : \begin{array}{|c|c|} \hline 100 & \text{---} \\ \hline & \text{ACAG} \\ \hline \end{array} \quad (3.19)$$

$$D_2 : \begin{array}{|c|c|} \hline 50 & \text{---} \\ \hline & \text{AGCG} \\ \hline \end{array} \quad (3.20)$$

las cuales se aparearán a los cabos que resulten al final del cómputo y que reportarán el resultado del mismo. La molécula D_1 reporta que la máquina se detuvo en el estado S_3 , el cual es un estado de aceptación, mientras que la molécula D_2 reporta que la máquina se detuvo en el estado S_4 .

Así, tenemos que el conjunto Γ contiene las *moléculas de transición*, el conjunto Δ las de *detección*, y el conjunto Σ el alfabeto *molecular*; y si hacemos $\Phi = \{FokI\}$, el conjunto de las enzimas de restricción, entonces la construcción $\langle \Gamma, \Delta, \Sigma, \Phi \rangle$ determina *completamente* nuestro DFA molecular.

Ahora, el proceso del DFA inicia al mezclar el hardware (enzima de restricción, *Ligasa* y *ATP*), el software (moléculas de transición) más la entrada, y el cómputo corre de manera autónoma hasta su terminación.

La manera en que se *arma* la entrada es la siguiente; esto es lo que se conocerá como *configuración inicial*.

...	RS	7	ENTRADA	TGTCGC ACAGCG	512	...
-----	----	---	---------	------------------	-----	-----

La idea es que esta cadena puede contener el número de pares que se quiera, pero sólo a partir de que se identifica el sitio de reconocimiento *RS* es donde comienza la entrada de la máquina y, por ende, el cómputo. De igual manera, la entrada, y el cómputo, termina después del carácter de terminación *t*, más allá de ese carácter puede haber cualquier par, pero la máquina *no lee* más de este byte. Es por eso que se debe ser cuidadoso de los pares que se coloquen en los espacios arbitrarios (en la configuración inicial debe cuidarse lo que se ponga en el lugar que ocupa el 7). Donde se lee *ENTRADA* en la configuración inicial es el lugar que ocupan los bytes que representan los símbolos 0 y 1; por ejemplo si queremos calcular $0 \text{ NAND } 1$ deberíamos armar la entrada como la siguiente cadena de ADN:

...	RS	7	CTGGCTAG GACCGATC	CGCAGCTA GCGTCGAT	TGTCGC ACAGCG	512	...
-----	----	---	----------------------	----------------------	------------------	-----	-----

(Cuya acción sobre esta máquina molecular se describió en la Introducción.)

Una vez expuesto lo anterior, estamos en condiciones de enunciar nuestra primera afirmación acerca de esta máquina molecular.

Afirmación 3.1.1. $\langle \Gamma, \Delta, \Sigma, \Phi \rangle$ resuelve $NAND_1$ con $n = 2$.

Demostración. La función $NAND_1$ para $n = 2$ es simplemente la tabla de verdad (3.1). Por tanto, para probar esta afirmación basta con mostrar que las moléculas en Γ realizan correctamente las transiciones descritas en el DFA que muestra la figura (3.1).

Hay que recordar que en el nivel de los autómatas finitos no existe así como una cinta sobre la cual se lee y se escribe (como sí se encuentra en las *máquinas de Turing*), en su lugar se debe interpretar el estado final del cómputo como un carácter de salida, de tal forma que cuando nuestro DFA se detiene en el estado S_3 decimos que la máquina *devuelve* un 1, pero si la máquina se detiene en el estado S_4 devuelve un 0.

Como acabamos de decir, se tiene que demostrar que en efecto las moléculas de transición en Γ , actuando sobre cadenas en Σ y Δ mediante la enzima en Φ , imitan perfectamente la acción descrita en la tabla (3.1).

- Por demostrar que $(0, 0) \mapsto 1$

Para demostrar esto hacemos la siguiente configuración inicial:

RS	7	CTGGCTAG GACCGATC	CTGGCTAG GACCGATC	TGTCGC ACAGCG	(3.21)
----	---	----------------------	----------------------	------------------	--------

El primer paso se da cuando la enzima *FokI* reconoce la cadena doble *GGATG* (abreviada por *RS* en la configuración anterior) y corta como le corresponde (ver arriba), dejando por un lado la cadena:

RS	7	CT GACCGA
----	---	--------------

y por el otro la cadena:

GGCTAG	CTGGCTAG GACCGATC	TGTCGC ACAGCG	(3.22)
TC			

Como describimos anteriormente en la cadena (3.16), el cabo pegajoso *GGCT* en esta segunda cadena indica que la máquina se encuentra en el estado S_0 con entrada 0.

En este punto, sólo una molécula del conjunto Γ es susceptible de aparearse con la cadena (3.22) dado el cabo pegajoso *GGCT* (en virtud de la *complementariedad* de las cadenas de ADN), a saber, aquella que tenga como cabo pegajoso la cadena inferior *CCGA*, o sea, la molécula de transición T_1 (3.8). (Recordemos que estamos descartando la utilización de moléculas que permiten este enlace, como son la *Ligasa* y el *ATP*, pues estamos considerando solamente la abstracción y no el experimento.)

Así, al unirse la molécula de transición (3.8) a la cadena (3.22), resulta la cadena:

RS	3	GGCTAG CCGATC	CTGGCTAG GACCGATC	TGTCGC ACAGCG	(3.23)
----	---	------------------	----------------------	------------------	--------

En esta cadena (3.23) se ha resaltado el cabo pegajoso *CCGA* en negritas por corresponder a la molécula (3.8). Comparemos la configuración inicial (3.21) con la cadena (3.23). Ambas cadenas son susceptibles de ser cortadas por la enzima *FokI*, en virtud de la presencia del sitio de reconocimiento *RS*; pero podemos mencionar las siguientes diferencias entre una cadena y otra:

1. El número de pares arbitrarios colocados a continuación del sitio de reconocimiento *RS*. La configuración inicial (3.21) cuenta con 7 pares-base arbitrarias, mientras que la cadena (3.23) cuenta sólo con 3; esto resulta en un corte distinto, que es precisamente lo que se pretende lograr.
2. La composición de la molécula, es decir, la combinación de pares que integran la cadena (3.23) difieren ligeramente de las de la configuración (3.21); lo que también resultará en el apareamiento de una molécula distinta a T_1 .

Nuevamente suponemos que la enzima *FokI* reconoce y corta la cadena (3.23), dejando 2 cadenas de las cuales la que nos importa es:



Cuyo cabo pegajoso es *CTGG*, el cual según describimos en (3.14) indica que la máquina se encuentra actualmente en el estado S_1 con entrada 0, lo cual es correcto como puede verificarse en el diagrama del DFA.

Análogamente, sólo existe una molécula en Γ susceptible de aparearse con la cadena (3.24) en virtud de su cabo pegajoso, a saber, la transición T_3 (3.10), ya que es la única molécula en Γ que posee el cabo pegajoso *GACC* en su cadena inferior.

De tal forma que al unirse ambas cadenas dan como resultado la cadena siguiente:



Esta cadena se ve más corta que la cadena (3.23), y naturalmente aun más que la configuración inicial (3.21); pero al igual que ellas, es susceptible de ser cortada por la enzima *FokI*, de tal manera que si suponemos que sucede así, la cadena restante es:



El cabo pegajoso *TGTC* indica que la máquina se encuentra en el estado S_3 con entrada t . Aquí sólo una molécula puede aparearse con la cadena (3.26), a saber una que no está en Γ , sino en Δ , la molécula de detección D_1 (3.19). De tal forma que al unirse ambas cadenas, en virtud de su cabo pegajoso, dan como resultado la siguiente cadena doble:



Esta cadena puede ser leída mediante secuenciación, o separada por electroforesis, y reporta que la máquina ha concluido el cómputo deteniéndose en el estado S_3 , el cual, como mencionamos arriba, es interpretado como un 1 de salida, lo que prueba que $(0, 0) \mapsto 1$.

Ahora pasamos a analizar otro caso:

- *P.d.* $(0, 1) \mapsto 1$

En esta situación tenemos la siguiente configuración inicial:

RS	7	CTGGCTAG	CGCAGCTA	TGTCGC
		GACCGATC	GCGTCGAT	ACAGCG

(3.28)

La enzima *FokI* reconoce y corta dejando la cadena:

GGCTAG	CGCAGCTA	TGTCGC
TC	GCGTCGAT	ACAGCG

(3.29)

Donde el cabo *GGCT* indica que la máquina se encuentra en el estado S_0 con entrada 0. En este punto, la molécula T_1 (3.8) se pega a la cadena anterior (3.29), dando como resultado la cadena doble:

RS	3	GGCTAG	CGCAGCTA	TGTCGC
		CCGATC	GCGTCGAT	ACAGGC

(3.30)

La cual es reconocida nuevamente por la enzima *FokI* que la corta dejando la cadena:

CGCAGCTA	TGTCGC
CGAT	ACAGGC

(3.31)

Cuyo cabo, *CGCA*, indica que el DFA se encuentra en el estado S_1 con entrada 1. En este punto, la molécula de transición T_4 (3.11) se pega con la cadena (3.31) en virtud de sus cabos. Por tanto, resulta la cadena:

RS	1	CGCAGCTA	TGTCGC
		GCGTCGAT	ACAGGC

(3.32)

De tal forma que al reconocer *FokI*, la cadena que resulta de su acción es:

$$\begin{array}{c} \text{TGTCGC} \\ \text{CG} \end{array}$$

que es igual a la cadena (3.26); y por tanto, podemos estar seguros que la molécula de detección D_1 (3.19) se apareará a ella, dando como resultado la cadena doble:

100	TGTCGC
	ACAGCG

tal como ocurrió con la cadena (3.27). Por lo tanto, la máquina se detuvo en el estado de aceptación S_3 como esperabamos.

- *P.d.* $(1, 0) \mapsto 1$

Tenemos la siguiente configuración inicial:

RS	7	CGCAGCTA	CTGGCTAG	TGTCGC
		GCGTCGAT	GACCGATC	ACAGCG

(3.33)

La enzima *FokI* reconoce y corta, dejando la cadena:



La cadena anterior (3.34) es reconocida por la molécula de transición T_2 (3.9) en virtud del cabo $CAGC$, el cual además indica que la máquina se encuentra en el estado S_0 con entrada 1. El resultado del apareamiento de las cadenas (3.34) y (3.9) es la cadena doble:



Nuevamente, la enzima *FokI* reconoce y corta, dejando la cadena:



El cabo $TGGC$ indica que la máquina se encuentra actualmente en el estado S_2 con entrada 0. En virtud de este cabo, la molécula de transición T_5 (3.12) se apareará a la cadena (3.36) dando como resultado la cadena doble:



La enzima *FokI* reconoce por última vez. Al cortar deja la siguiente cadena:



, la cual es igual a la cadena (3.26). Por lo tanto, podemos asegurar que la molécula de detección D_1 (3.19) se apareará a ella en virtud del cabo $TGTC$, dando como resultado la cadena (3.27).



Esto implica que $(1,0) \mapsto 1$. Resta verificar un último caso:

- *P.d.* $(1,1) \mapsto 0$

Tenemos la siguiente configuración inicial:



La enzima *FokI* reconoce y corta:



De tal forma que el cabo $CAGC$ indica que el DFA se encuentra en el estado S_0 con entrada 1. En este punto sólo una molécula es susceptible de aparearse con la cadena (3.40); a saber, la molécula de transición T_2 (3.9). Esta molécula al pegarse con (3.40) forma la cadena doble:

RS	2	CAGCTA GTCGAT	CGCAGCTA GCGTCGAT	TGTCGC ACAGCG
----	---	-------------------------	----------------------	------------------

(3.41)

Al reconocer nuevamente la enzima *FokI* corta la cadena anterior (3.41) dejando la cadena:

GCAGCTA	TGTCGC
GAT	ACAGCG

(3.42)

Cuyo cabo *GCAG* indica que la máquina se encuentra actualmente en el estado S_2 con entrada 1. A este cabo se pega la molécula T_6 (3.13) dejando como resultado la siguiente cadena:

RS	0	GCAGCTA CGTCGAT	TGTCGC ACAGCG
----	---	---------------------------	------------------

(3.43)

El cual es cortado por la enzima *FokI* resultando en la siguiente cadena:

TCGC
—

(3.44)

Cuyo cabo *TCGC* se adhiere a la molécula de detección D_2 (3.20); dando como resultado la siguiente cadena:

50	TCGC AGCG
----	---------------------

(3.45)

Recordemos que esta molécula reporta que el cómputo finalizó con el estado S_4 que es interpretado como un 0 de salida. Lo cual prueba que $(1, 1) \mapsto 0$, como deseábamos. □

La prueba a la afirmación (3.1.1) sirve para exhibir la forma en que el cómputo con ADN se lleva a cabo¹. También se puede ver la importancia de las moléculas de transición y de detección, ya que en ellas reside la esencia del cómputo molecular. Por lo tanto, su diseño debe de tomar en cuenta las *enzimas de restricción* que se vayan a emplear, así como también la *longitud* de las cadenas de entrada (la longitud de palabra) y la selección de bases para las mismas. Ya teniendo claro la forma en que estos elementos interactúan, estamos en condiciones de generalizar el autómata que resuelve la función $NAND_1$.

3.2. $NAND_1$ con $n \geq 2$

Ahora analizaremos la generalización de la *1a Aproximación*, $NAND_1$, i.e. cuando $n \geq 2$. El autómata que resuelve esta generalización es el que está descrito en el siguiente diagrama, figura (3.3). Como ya habíamos comentado en el capítulo anterior, el proceso se lleva a cabo sobre una cadena de 0s y 1s, leyendo la cadena de izquierda a derecha y calculando la función NAND de una pareja de bits, depositando el resultado en la posición que ocupa el bit derecho.

¹Asumiendo un cómputo serial, es decir, descartando la acción que otras moléculas puedan ejercer en cada uno de los pasos intermedios del proceso.

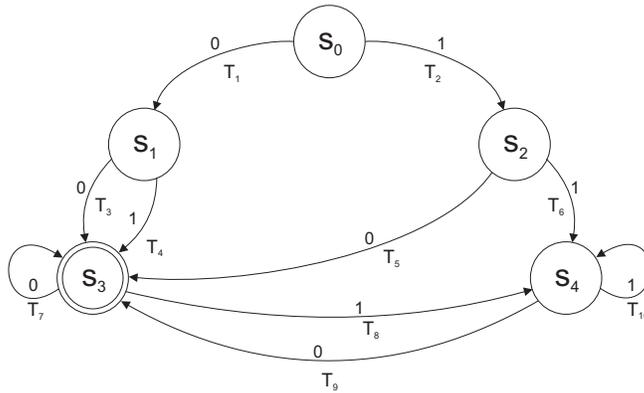


Figura 3.3: Autómata Determinista Finito

Por ejemplo, para $n = 10$, es decir, para $NAND_1 : \{0, 1\}^{10} \rightarrow \{0, 1\}$, tenemos que

$$(1, 0, 0, 1, 0, 1, 1, 1, 0, 1) \mapsto 0$$

Ya que:

```

1001011101
101011101
11011101
0011101
111101
01101
1101
001
11
0

```

Del autómata anterior, descrito en la figura (3.3), se siguen las siguientes transiciones:

$$\begin{aligned}
 T_1 : S_0 &\xrightarrow{0} S_1 \\
 T_2 : S_0 &\xrightarrow{1} S_2 \\
 T_3 : S_1 &\xrightarrow{0} S_3 \\
 T_4 : S_1 &\xrightarrow{1} S_3 \\
 T_5 : S_2 &\xrightarrow{0} S_3 \\
 T_6 : S_2 &\xrightarrow{1} S_4 \\
 T_7 : S_3 &\xrightarrow{0} S_3
 \end{aligned} \tag{3.46}$$

$$T_8 : S_3 \xrightarrow{1} S_4 \quad (3.47)$$

$$T_9 : S_4 \xrightarrow{0} S_3 \quad (3.48)$$

$$T_{10} : S_4 \xrightarrow{1} S_4 \quad (3.49)$$

De las cuales, las primeras 6 son iguales al autómata de la sección anterior, y los últimos 4 corresponden a transiciones que logran la generalización del autómata.

Nuestro modelo anterior $\langle \Gamma, \Delta, \Sigma, \Phi \rangle$ fue sencillo y resuelve una aplicación sencilla; ahora, para la generalización de la *1a Aproximación* podemos refinar nuestro modelo y considerar moléculas de detección que no sólo detecten si la máquina se detuvo en el estado S_3 o S_4 , sino que también reporten si el cómputo terminó en un estado intermedio (S_0, S_1 o S_2).

Para esto, simplifiquemos más la representación de nuestras cadenas. De tal forma que el símbolo 0 será representado molecularmente por la cadena:

$$0 = \boxed{A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8} \quad (3.50)$$

para algunos $A_i \in \{A, C, G, T\}$. Se adoptó una convención más: sólo se hará referencia a la cadena complementaria cuando sea necesario (e.g. tras la acción de una enzima, para mostrar el cabo pegajoso).

Así, la cadena (3.50) es la versión abreviada de la cadena:

$$\boxed{\begin{array}{c} \widetilde{A_1} \widetilde{A_2} \widetilde{A_3} \widetilde{A_4} \widetilde{A_5} \widetilde{A_6} \widetilde{A_7} \widetilde{A_8} \\ \widetilde{A_1} \widetilde{A_2} \widetilde{A_3} \widetilde{A_4} \widetilde{A_5} \widetilde{A_6} \widetilde{A_7} \widetilde{A_8} \end{array}}$$

en donde $\widetilde{A_i}$ es la base complementaria de A_i ; por ejemplo, si $A_i = C$, entonces $\widetilde{A_i} = G$.

El símbolo 1 queda representado molecularmente como:

$$1 = \boxed{B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8} \quad (3.51)$$

para algunos $B_i \in \{A, C, G, T\}$.

Y en esta ocasión el carácter de terminación t , tiene 8 bits como longitud de palabra, igual que los símbolos 1 y 0. Este símbolo queda representado molecularmente así:

$$t = \boxed{C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8} \quad (3.52)$$

para algunos $C_i \in \{A, C, G, T\}$.

Naturalmente, la selección que hagamos para las bases que integran los caracteres 0, 1 y t deben diferir al menos en una base. Para éstas (ya sea 0, 1 ó t) la codificación de los 5 estados (S_0, S_1, S_2, S_3 y S_4) sobre los cabos pegajosos de la cadena N queda como sigue:

$$N = \begin{array}{c} N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 \\ \widetilde{N_1} \widetilde{N_2} \widetilde{N_3} \widetilde{N_4} \widetilde{N_5} \widetilde{N_6} \widetilde{N_7} \widetilde{N_8} \end{array}$$

1. Cuando el cabo representa el estado S_4 la cadena se ve:

$$N_1 N_2 N_3 N_4 \overline{N_5 N_6 N_7 N_8} \quad (3.53)$$

2. Cuando el cabo representa el estado S_3 la cadena se ve:

$$N_2 N_3 N_4 N_5 \overline{N_6 N_7 N_8} \quad (3.54)$$

3. Cuando el cabo representa el estado S_2 la cadena se ve:

$$N_3 N_4 N_5 N_6 \overline{N_7 N_8} \quad (3.55)$$

4. Cuando el cabo representa el estado S_1 la cadena se ve:

$$N_4 N_5 N_6 N_7 \overline{N_8} \quad (3.56)$$

5. Cuando el cabo representa el estado S_0 la cadena se ve:

$$\overline{N_5 N_6 N_7 N_8} \quad (3.57)$$

Así, por ejemplo, la cadena

$$A_1 A_2 A_3 A_4 \overline{A_5 A_6 A_7 A_8}$$

indica que la máquina molecular se encuentra en el estado S_4 con entrada 0.

Ahora pasemos a describir la versión molecular de las transiciones de este autómata, figura (3.3). Podemos esperar que estas moléculas difieran de las moléculas de transición empleadas en la sección pasada, ya que no sólo tenemos 4 transiciones adicionales, sino que hemos cambiado la estructura del símbolo de terminación t y no hemos hecho una asignación de las bases que formarán A_i , B_i y C_i .

Nuestro nuevo conjunto Γ queda integrado como sigue:

$$T_1 : \begin{array}{|c|c|c|} \hline \text{RS} & 2 & \overline{A_5 A_6 A_7 A_8} \\ \hline \end{array} \quad (3.58)$$

$$T_2 : \begin{array}{|c|c|c|} \hline \text{RS} & 3 & \overline{B_5 B_6 B_7 B_8} \\ \hline \end{array} \quad (3.59)$$

$$T_3 : \begin{array}{|c|c|c|} \hline \text{RS} & 3 & \overline{A_4 A_5 A_6 A_7} \\ \hline \end{array} \quad (3.60)$$

$$T_4 : \begin{array}{|c|c|c|} \hline \text{RS} & 3 & \overline{B_4 B_5 B_6 B_7} \\ \hline \end{array} \quad (3.61)$$

$$T_5 : \begin{array}{|c|c|c|} \hline \text{RS} & 2 & \widetilde{\widetilde{\widetilde{\widetilde{A_3 A_4 A_5 A_6}}}} \\ \hline \end{array} \quad (3.62)$$

$$T_6 : \begin{array}{|c|c|c|} \hline \text{RS} & 3 & \widetilde{\widetilde{\widetilde{\widetilde{B_3 B_4 B_5 B_6}}}} \\ \hline \end{array} \quad (3.63)$$

$$T_7 : \begin{array}{|c|c|c|} \hline \text{RS} & 1 & \widetilde{\widetilde{\widetilde{\widetilde{A_2 A_3 A_4 A_5}}}} \\ \hline \end{array} \quad (3.64)$$

$$T_8 : \begin{array}{|c|c|c|} \hline \text{RS} & 2 & \widetilde{\widetilde{\widetilde{\widetilde{B_2 B_3 B_4 B_5}}}} \\ \hline \end{array} \quad (3.65)$$

$$T_9 : \begin{array}{|c|c|c|} \hline \text{RS} & 0 & \widetilde{\widetilde{\widetilde{\widetilde{A_1 A_2 A_3 A_4}}}} \\ \hline \end{array} \quad (3.66)$$

$$T_{10} : \begin{array}{|c|c|c|} \hline \text{RS} & 1 & \widetilde{\widetilde{\widetilde{\widetilde{B_1 B_2 B_3 B_4}}}} \\ \hline \end{array} \quad (3.67)$$

y las moléculas de detección que integran el conjunto Δ son las siguientes:

$$D_1 : \begin{array}{|c|c|} \hline r & \widetilde{\widetilde{\widetilde{\widetilde{C_5 C_6 C_7 C_8}}}} \\ \hline \end{array} \quad (3.68)$$

$$D_2 : \begin{array}{|c|c|} \hline x & \widetilde{\widetilde{\widetilde{\widetilde{C_4 C_5 C_6 C_7}}}} \\ \hline \end{array} \quad (3.69)$$

$$D_3 : \begin{array}{|c|c|} \hline y & \widetilde{\widetilde{\widetilde{\widetilde{C_3 C_4 C_5 C_6}}}} \\ \hline \end{array} \quad (3.70)$$

$$D_4 : \begin{array}{|c|c|} \hline z & \widetilde{\widetilde{\widetilde{\widetilde{C_2 C_3 C_4 C_5}}}} \\ \hline \end{array} \quad (3.71)$$

$$D_5 : \begin{array}{|c|c|} \hline w & \widetilde{\widetilde{\widetilde{\widetilde{C_1 C_2 C_3 C_4}}}} \\ \hline \end{array} \quad (3.72)$$

con $r, x, y, z, w \in \mathbb{N}$ que representan la longitud de la cadena de pares que debe colocarse en esa posición. Hay que recordar que queremos extender la potencia de nuestro DFA, haciendo posible que reporte el fin de un cómputo en estados distintos de S_3 y S_4 ; por esa razón, hemos implementado cinco moléculas de detección en comparación a las dos del DFA de la sección anterior. De esta manera, D_1 sirve para detectar que la máquina se detuvo en el estado S_0 , D_2 para detectar paro en el estado S_1 , D_3 para detectar paro en el estado S_2 , D_4 para reportar que la máquina se detuvo en el estado S_3 y por último, D_5 nos indica que la máquina se detuvo en el estado S_4 .

Como en el caso de la sección anterior, debemos asegurarnos que esta nueva máquina (definida por $\Gamma = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}\}$, $\Delta = \{D_1, D_2, D_3, D_4, D_5\}$, $\Sigma = \{0, 1, t\}$ y $\Phi = \{FokI\}$) resuelve la función $NAND_1$ para $n \geq 2$. Y esto es lo que probaremos en el siguiente teorema.

Teorema 3.2.1. $\langle \Gamma, \Delta, \Sigma, \Phi \rangle$ resuelve $NAND_1$ con $k \geq 2$

Demostración. Recordemos el autómata que queremos simular mediante moléculas de ADN, figura (3.3).

Supongamos la siguiente cadena:

$$\boxed{\text{RS} \quad 5 \quad \lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_k \quad \cdots} \quad (3.73)$$

donde $\lambda_i \in \Sigma$; y por tanto la longitud de cada λ_i es de 8 pares.

Adoptaremos la siguiente convención.

$$\boxed{8_{\lambda_i}} \quad (3.74)$$

El bloque (3.74) denota una cadena de 8 pares correspondientes al símbolo λ_i . De tal forma que la cadena (3.73) puede escribirse de la siguiente manera:

$$\boxed{\text{RS} \quad 5 \quad 8_{\lambda_1} \quad 8_{\lambda_2} \quad \cdots \quad 8_{\lambda_k} \quad \cdots} \quad (3.75)$$

La cadena (3.75) corresponde a la configuración inicial de esta máquina.

Esta prueba supondrá varios valores de n , es decir, no se restringe a una sola $n \geq 2$ para $NAND_1 : \{0, 1\}^n \rightarrow \{0, 1\}$; sino que asumirá distintos valores para esta variable y que se manifestará en el número de bytes en la entrada de la máquina. Para realizar esta prueba basta demostrar que los elementos de Γ y Δ se aplican correctamente a cada una de las instancias que tendrá la cadena (3.75).

También es necesario mencionar que esta prueba es recurrente, pues se verá que conforme se alimenta la máquina de bytes de entrada, el DFA caerá en casos que ya habían sido descritos.

El proceso inicia cuando la enzima *FokI* reconoce y corta la cadena (3.75) resultando en la siguiente cadena:

$$\lambda_5^1 \lambda_6^1 \lambda_7^1 \lambda_8^1 \boxed{8_{\lambda_2} \quad 8_{\lambda_3} \quad \cdots} \quad (3.76)$$

Como se dijo arriba, las bases del cabo pegajoso codifican el estado actual de la máquina; en este caso, el cabo $\lambda_5^1 \lambda_6^1 \lambda_7^1 \lambda_8^1$ (el índice superior indica el número del byte y el inferior el número de base) reporta que el DFA está en el estado S_0 con entrada λ_1 . Resulta la siguiente casuística cuando nos preguntamos por el valor de λ_1 :

CASO 1: $\lambda_1 = t$ Es decir, la entrada de la máquina es únicamente el carácter de terminación t . Entonces la cadena (3.76) en realidad es la cadena:

$$C_5 C_6 C_7 C_8 \boxed{8_{\lambda_2} \quad 8_{\lambda_3} \quad \cdots} \quad (3.77)$$

En este caso, sólo una cadena es susceptible de aparearse con la cadena (3.77) en virtud de su cabo; a saber, la molécula de detección D_1 (3.68). Una vez que esto sucede, la cadena se ve así:

$$\boxed{r \quad \begin{array}{c} \widetilde{C_5 C_6 C_7 C_8} \\ \widetilde{C_5 C_6 C_7 C_8} \end{array} \quad 8_{\lambda_2} \quad \cdots}$$

Y por tanto, el proceso concluye reportándonos que la máquina se detuvo en el estado S_0 .

Observación 3.2.1. *En este caso, $n = 0$ pues no hubo una entrada per se. Mencionamos que para esta aproximación debe cumplirse que $n \geq 2$, pero lo anterior sirve para exhibir cómo la máquina está preparada para enfrentar estos casos.*

CASO 2: $\lambda_1 = 0$ Es decir, el primer carácter de entrada es 0. Entonces la cadena (3.76) en realidad es la cadena:

$$A_5 A_6 A_7 A_8 \begin{array}{|c|c|c|} \hline \delta_{\lambda_2} & \delta_{\lambda_3} & \dots \\ \hline \end{array} \quad (3.78)$$

Según el diagrama del DFA, esperamos que la molécula de transición T_1 (3.58) sea la que se aparee a la cadena anterior (3.78); lo cual efectivamente ocurre dado el cabo $A_5 A_6 A_7 A_8$. Así obtenemos la siguiente cadena:

$$\begin{array}{|c|c|c|c|c|} \hline \text{RS} & 2 & \begin{array}{c} \widetilde{A_5 A_6 A_7 A_8} \\ \widetilde{A_5 A_6 A_7 A_8} \end{array} & \delta_{\lambda_2} & \dots \\ \hline \end{array} \quad (3.79)$$

CASO 3: $\lambda_1 = 1$ Es decir, el primer carácter de entrada es 1. Entonces la cadena (3.76) en realidad es la cadena:

$$B_5 B_6 B_7 B_8 \begin{array}{|c|c|c|} \hline \delta_{\lambda_2} & \delta_{\lambda_3} & \dots \\ \hline \end{array} \quad (3.80)$$

Según el diagrama del DFA, esperamos que la molécula de transición T_2 (3.59) sea la que se aparee a la cadena anterior (3.80); lo cual efectivamente ocurre dado el cabo $B_5 B_6 B_7 B_8$. Así obtenemos la siguiente cadena:

$$\begin{array}{|c|c|c|c|c|} \hline \text{RS} & 3 & \begin{array}{c} \widetilde{B_5 B_6 B_7 B_8} \\ \widetilde{B_5 B_6 B_7 B_8} \end{array} & \delta_{\lambda_2} & \dots \\ \hline \end{array} \quad (3.81)$$

Observación 3.2.2. *En los dos casos anteriores (CASO 2 y CASO 3) estamos suponiendo que $n \geq 1$ pues al menos hay un carácter de entrada.*

Ahora procedemos a desarrollar estos casos (naturalmente, no desarrollamos el **CASO 1**, ya que la máquina se ha detenido en esta situación). Comenzamos desarrollando el **CASO 2**:

Suponemos la cadena (3.79) y hasta ahora hemos leído el carácter 0. La enzima *FokI* reconoce y corta, dejando la cadena:

$$\begin{array}{c} \lambda_4^2 \lambda_5^2 \lambda_6^2 \lambda_7^2 \lambda_8^2 \\ \lambda_8^2 \end{array} \begin{array}{|c|c|c|} \hline \delta_{\lambda_3} & \delta_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.82)$$

Puede observarse que el cabo $\lambda_4^2 \lambda_5^2 \lambda_6^2 \lambda_7^2 \lambda_8^2$ de la cadena (3.82) indica que la máquina se encuentra en el estado S_1 con entrada λ_2 . Cuando nos preguntamos sobre el valor de λ_2 resulta la siguiente casuística:

CASO 2.1: $\lambda_2 = t$ Es decir, la entrada de la máquina es únicamente el carácter 0 más el carácter de terminación t . Entonces la cadena (3.82) en realidad es la cadena:

$$C_4C_5C_6C_7C_8 \begin{array}{|c|c|c|} \hline 8_{\lambda_3} & 8_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.83)$$

En este caso, la máquina debe detenerse en el estado S_1 . Sólo una cadena es susceptible de aparearse con la cadena (3.83) en virtud de su cabo; a saber, la molécula de detección D_2 (3.69). Una vez que esto sucede, la cadena se ve así:

$$x \begin{array}{|c|c|c|} \hline \widetilde{C_4C_5C_6C_7C_8} & 8_{\lambda_3} & \dots \\ \hline \end{array}$$

Y por tanto, el proceso concluye reportándonos que la máquina se detuvo en el estado S_1 .

Observación 3.2.3. *En este caso, $n = 1$ lo cual no puede ocurrir si hemos definido la función para $n \geq 2$, pero aquí se exhibe este caso para mostrar cómo la máquina hace frente a estas situaciones.*

CASO 2.2: $\lambda_2 = 0$ Es decir, hasta el momento se ha leído la cadena 00. Así, la cadena (3.82) en realidad es la cadena:

$$A_4A_5A_6A_7A_8 \begin{array}{|c|c|c|} \hline 8_{\lambda_3} & 8_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.84)$$

Aquí, la única molécula que se aparea al cabo de la cadena (3.84) es la molécula de transición T_3 (3.60), que es lo esperado según el diagrama del DFA. Así obtenemos la siguiente cadena:

$$\text{RS} \begin{array}{|c|c|c|c|} \hline 3 & \widetilde{A_4A_5A_6A_7A_8} & 8_{\lambda_3} & \dots \\ \hline \end{array} \quad (3.85)$$

CASO 2.3: $\lambda_2 = 1$ Es decir, hasta el momento se han leído los caracteres 01, siendo el carácter 1 el que se lee actualmente. Entonces la cadena (3.82) en realidad es la cadena:

$$B_4B_5B_6B_7B_8 \begin{array}{|c|c|c|} \hline 8_{\lambda_3} & 8_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.86)$$

Según el diagrama del DFA, esperamos que la molécula de transición T_4 (3.61) sea la que se aparee a la cadena anterior (3.86); lo cual efectivamente ocurre dado el cabo $B_4B_5B_6B_7$. Así obtenemos la siguiente cadena:

$$\text{RS} \begin{array}{|c|c|c|c|} \hline 3 & \widetilde{B_4B_5B_6B_7B_8} & 8_{\lambda_3} & \dots \\ \hline \end{array} \quad (3.87)$$

Hemos llegado a un punto en el cual debemos desarrollar uno de los dos casos anteriores. Los dos últimos casos nos hacen ver que estamos considerando $n \geq 2$, lo cual vale para la función $NAND_1$ que queremos simular mediante moléculas de ADN. Ahora desarrollaremos el **CASO 2.2**, pues el **CASO 2.1** carece de sentido ya que la máquina se ha detenido en ese caso.

Tomemos la cadena (3.85). Hasta el momento se ha leído la cadena 00 y por tanto, esperaríamos estar en el estado S_3 , lo cual se verificará tras la acción de la enzima *FokI*. Esta enzima reconoce y corta, dejando la siguiente cadena:

$$\lambda_2^3 \lambda_3^3 \lambda_4^3 \lambda_5^3 \lambda_6^3 \lambda_7^3 \lambda_8^3 \begin{array}{|c|c|c|} \hline \lambda_6^3 \lambda_7^3 \lambda_8^3 & 8_{\lambda_4} & 8_{\lambda_5} & \dots \\ \hline \end{array} \quad (3.88)$$

Por (3.54) sabemos que el cabo $\lambda_2^3 \lambda_3^3 \lambda_4^3 \lambda_5^3$ indica que la máquina se encuentra en el estado S_3 con entrada λ_3 . Ahora analicemos la casuística para el byte λ_3 .

CASO 2.2.1: $\lambda_3 = t$ Es decir, la entrada consiste en la cadena 00 más el carácter de terminación t . Entonces la cadena (3.88) en realidad es la cadena:

$$C_2 C_3 C_4 C_5 \underbrace{C_6 C_7 C_8}_{\widetilde{C_6 \widetilde{C_7} \widetilde{C_8}}} \begin{array}{|c|c|c|} \hline 8_{\lambda_4} & 8_{\lambda_5} & \dots \\ \hline \end{array} \quad (3.89)$$

Aquí, la máquina debe detenerse en el estado S_3 , lo cual ocurre efectivamente ya que sólo una cadena es susceptible de aparearse con la cadena (3.89) en virtud de su cabo; esta es la molécula de detección D_4 (3.71). Una vez que esto sucede, la cadena se ve así:

$$\begin{array}{|c|c|c|} \hline z & \underbrace{C_2 C_3 C_4 C_5 C_6 C_7 C_8}_{\widetilde{C_2 \widetilde{C_3} \widetilde{C_4} \widetilde{C_5} \widetilde{C_6} \widetilde{C_7} \widetilde{C_8}}} & 8_{\lambda_4} & \dots \\ \hline \end{array}$$

Y por tanto, el proceso concluye reportándonos que la máquina se detuvo en el estado S_3 .

CASO 2.2.2: $\lambda_3 = 0$ Es decir, hasta el momento se ha leído la cadena 000. Así, la cadena (3.88) en realidad es la cadena:

$$A_2 A_3 A_4 A_5 \underbrace{A_6 A_7 A_8}_{\widetilde{A_6 \widetilde{A_7} \widetilde{A_8}}} \begin{array}{|c|c|c|} \hline 8_{\lambda_4} & 8_{\lambda_5} & \dots \\ \hline \end{array} \quad (3.90)$$

Aquí, la única molécula que se aparea al cabo de la cadena (3.90) es la molécula de transición T_7 (3.64), que es lo esperado según el diagrama del DFA. Así obtenemos la siguiente cadena:

$$\begin{array}{|c|c|c|} \hline \text{RS} & 1 & \underbrace{A_2 A_3 A_4 A_5 A_6 A_7 A_8}_{\widetilde{A_2 \widetilde{A_3} \widetilde{A_4} \widetilde{A_5} \widetilde{A_6} \widetilde{A_7} \widetilde{A_8}}} & 8_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.91)$$

CASO 2.2.3: $\lambda_3 = 1$ Es decir, hasta el momento se han leído los caracteres 001, siendo el carácter 1 el que se lee actualmente. Entonces la cadena (3.88) en realidad es la cadena:

$$B_2 B_3 B_4 B_5 \underbrace{B_6 B_7 B_8}_{\widetilde{B_6 \widetilde{B_7} \widetilde{B_8}}} \begin{array}{|c|c|c|} \hline 8_{\lambda_4} & 8_{\lambda_5} & \dots \\ \hline \end{array} \quad (3.92)$$

Según el diagrama del DFA, esperamos que la molécula de transición T_8 (3.65) sea la que se aparea a la cadena anterior (3.92); lo cual ocurre efectivamente. Así obtenemos la siguiente cadena:

$$\begin{array}{|c|c|c|} \hline \text{RS} & 2 & \underbrace{B_2 B_3 B_4 B_5 B_6 B_7 B_8}_{\widetilde{B_2 \widetilde{B_3} \widetilde{B_4} \widetilde{B_5} \widetilde{B_6} \widetilde{B_7} \widetilde{B_8}}} & 8_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.93)$$

Como puede verse, en los casos anteriores la cadena de entrada es de longitud $n = 3$. Supongamos a continuación que la cadena de entrada es mayor a 3 y desarrollemos los casos **2.2.2** y **2.2.3**. Empezamos por el **CASO 2.2.2** suponiendo la cadena (3.91). Recordemos que hasta este punto la entrada a la máquina ha sido la cadena 000.

Una vez que la enzima *FokI* reconoce y corta, obtenemos la cadena:

$$\lambda_2^4 \lambda_3^4 \lambda_4^4 \lambda_5^4 \lambda_6^4 \lambda_7^4 \lambda_8^4 \begin{array}{|c|c|c|} \hline 8_{\lambda_5} & 8_{\lambda_6} & \dots \\ \hline \end{array} \quad (3.94)$$

De tal forma que el cabo $\lambda_2^4 \lambda_3^4 \lambda_4^4 \lambda_5^4$ nos indica que estamos en el estado S_3 con entrada λ_4 .

Puede verse que la cadena (3.94) es semejante a la cadena (3.88), por lo cual podemos esperar la misma casuística que aquella cadena; a saber, alguno de los siguientes casos:

1. Si $\lambda_4 = t$, entonces la molécula de detección D_4 se pega a la cadena (3.94) y la máquina se detiene en S_3 con entrada 000.
2. Si $\lambda_4 = 0$, entonces la molécula de transición T_7 (3.64) se pega a (3.94). Esta transición corresponde a la entrada 0000.
3. Si $\lambda_4 = 1$, entonces la molécula de transición T_8 (3.65) se pega a la cadena (3.94) tal y como se espera para la entrada 0001.

En este punto hemos llevado el **CASO 2.2.2** hasta su último extremo, agotándolo para cualquier tamaño de entrada. Desarrollando otro **CASO** podremos concluir esta prueba.

Desarrollemos ahora el **CASO 2.2.3**. Trabajemos ahora con la cadena (3.93); y recordemos que la entrada hasta este punto es 001. Una vez que la enzima *FokI* reconoce y corta, obtenemos la cadena:

$$\lambda_1^4 \lambda_2^4 \lambda_3^4 \lambda_4^4 \lambda_5^4 \lambda_6^4 \lambda_7^4 \lambda_8^4 \begin{array}{|c|c|c|} \hline 8_{\lambda_5} & 8_{\lambda_6} & \dots \\ \hline \end{array} \quad (3.95)$$

De tal forma que el cabo $\lambda_1^4 \lambda_2^4 \lambda_3^4 \lambda_4^4$ nos indica que estamos en el estado S_4 con entrada λ_4 . Analicemos la casuística de λ_4 .

CASO 2.2.3.1: $\lambda_4 = t$ Es decir, la entrada consiste en la cadena 001 más el carácter de terminación t . Entonces la cadena (3.95) en realidad es la cadena:

$$C_1 C_2 C_3 C_4 \begin{array}{|c|c|c|} \hline 8_{\lambda_5} & 8_{\lambda_6} & \dots \\ \hline \end{array} \quad (3.96)$$

El cabo $C_1 C_2 C_3 C_4$ de la cadena (3.96) sólo puede unirse a la molécula de detección D_5 (3.72), obteniendo así:

$$w \begin{array}{|c|c|c|} \hline \frac{C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8}{\widetilde{C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8}} & 8_{\lambda_5} & \dots \\ \hline \end{array}$$

Y por tanto, el proceso concluye reportándonos que la máquina se detuvo en el estado S_4 como era esperado.

CASO 2.2.3.2: $\lambda_4 = 0$ Es decir, hasta el momento se ha leído la cadena 0010. Leyéndose actualmente el carácter 0. Así, la cadena (3.95) en realidad es la cadena:

$$\begin{array}{c} A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 \\ \widetilde{A_5 A_6 A_7 A_8} \end{array} \begin{array}{|c|c|c|} \hline 8_{\lambda_5} & 8_{\lambda_6} & \dots \\ \hline \end{array} \quad (3.97)$$

Aquí, la única molécula que se aparea al cabo de la cadena (3.97) es la molécula de transición T_9 (3.66), que es lo esperado según el diagrama del DFA. Así obtenemos la siguiente cadena:

$$\begin{array}{|c|c|c|c|c|} \hline \text{RS} & 0 & \begin{array}{c} A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 \\ \widetilde{A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8} \end{array} & 8_{\lambda_5} & \dots \\ \hline \end{array} \quad (3.98)$$

CASO 2.2.3.3: $\lambda_4 = 1$ Es decir, hasta el momento se han leído los caracteres 0011. Entonces la cadena (3.95) en realidad es la cadena:

$$\begin{array}{c} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 \\ \widetilde{B_5 B_6 B_7 B_8} \end{array} \begin{array}{|c|c|c|} \hline 8_{\lambda_5} & 8_{\lambda_6} & \dots \\ \hline \end{array} \quad (3.99)$$

Según el diagrama del DFA, esperamos que la molécula de transición T_{10} (3.67) sea la que se aparece a la cadena anterior (3.99); lo cual ocurre efectivamente. Así obtenemos la siguiente cadena:

$$\begin{array}{|c|c|c|c|c|} \hline \text{RS} & 1 & \begin{array}{c} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 \\ \widetilde{B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8} \end{array} & 8_{\lambda_5} & \dots \\ \hline \end{array} \quad (3.100)$$

Nuevamente, desarrollaremos los **CASOS 2.2.3.2** y **2.2.3.3** para completar la prueba. Comenzamos desarrollando el **CASO 2.2.3.2**. Tomemos la cadena (3.98) y recordemos que hasta este punto la cadena de entrada del DFA contiene 0010. Una vez que la enzima *FokI* reconoce y corta la cadena (3.98) tenemos:

$$\begin{array}{c} \lambda_2^5 \lambda_3^5 \lambda_4^5 \lambda_5^5 \lambda_6^5 \lambda_7^5 \lambda_8^5 \\ \widetilde{\lambda_6^5 \lambda_7^5 \lambda_8^5} \end{array} \begin{array}{|c|c|c|} \hline 8_{\lambda_6} & 8_{\lambda_7} & \dots \\ \hline \end{array} \quad (3.101)$$

De tal forma que el cabo $\lambda_2^5 \lambda_3^5 \lambda_4^5 \lambda_5^5$ nos indica que la máquina se encuentra actualmente en el estado S_3 con entrada λ_5 ; y como puede verse, la cadena (3.101) es análoga a la cadena (3.88), por lo cual obtenemos exactamente la misma casuística en este punto, entonces hemos agotado todas las posibilidades desarrollando el **CASO 2.2.3.2**. Desarrollemos ahora el **CASO 2.2.3.3**. Consideramos ahora la cadena (3.100), recordando que hasta este momento hemos leído la entrada 0011. Tras reconocer y cortar, la enzima *FokI* deja la siguiente cadena:

$$\begin{array}{c} \lambda_1^5 \lambda_2^5 \lambda_3^5 \lambda_4^5 \lambda_5^5 \lambda_6^5 \lambda_7^5 \lambda_8^5 \\ \widetilde{\lambda_5^5 \lambda_6^5 \lambda_7^5 \lambda_8^5} \end{array} \begin{array}{|c|c|c|} \hline 8_{\lambda_6} & 8_{\lambda_7} & \dots \\ \hline \end{array} \quad (3.102)$$

El cabo $\lambda_1^5 \lambda_2^5 \lambda_3^5 \lambda_4^5$ reporta que la máquina se encuentra en el estado S_4 con entrada λ_5 . Como hace un momento, podemos observar que la cadena (3.102) es similar a la cadena (3.95) y por tanto, el **CASO 2.2.3.3** resulta en la misma casuística que el **CASO 2.2.3**; a saber:

1. Si $\lambda_5 = t$, la molécula de detección D_5 (3.72) reconoce y se pega a la cadena (3.102), reportando que la máquina se detuvo en el estado S_4 .

2. Si $\lambda_5 = 0$, la entrada consiste, hasta el momento, de la cadena 00110, y la molécula T_9 (3.66) se adhiere a la cadena (3.102).
3. Si $\lambda_5 = 1$, la entrada consiste, hasta el momento, de la cadena 00111, y la molécula T_{10} (3.67) se adhiere a la cadena (3.102).

Como se puede ver, ya se analizó exhaustivamente todos los distintos casos que resultan de suponer la cadena (3.79) del **CASO 2**.

Analizaremos la última rama del DFA; aquella que resulta de desarrollar el **CASO 3**. Consideremos la cadena (3.81) y recordemos que en este punto sólo se ha leído el carácter 1 y por ende, esperaríamos estar en el estado S_2 .

Cuando la enzima *FokI* reconoce y corta la cadena (3.81), deja como resultado la cadena:

$$\lambda_3^2 \lambda_4^2 \lambda_5^2 \lambda_6^2 \lambda_7^2 \lambda_8^2 \begin{array}{|c|c|c|} \hline \lambda_7^2 \lambda_8^2 & 8_{\lambda_3} & 8_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.103)$$

El cabo $\lambda_3^2 \lambda_4^2 \lambda_5^2 \lambda_6^2$ nos indica que el DFA se encuentra en el estado S_2 con entrada λ_2 , como se puede deducir de (3.55). Analicemos la casuística de λ_2 .

CASO 3.1: $\lambda_2 = t$ Es decir, la entrada de la máquina es únicamente el carácter 1 más el carácter de terminación t . Entonces la cadena (3.103) en realidad es la cadena:

$$C_3 C_4 C_5 C_6 C_7 C_8 \begin{array}{|c|c|c|} \hline \widetilde{C_7 C_8} & 8_{\lambda_3} & 8_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.104)$$

En este caso, la máquina debe detenerse en el estado S_2 . Sólo una cadena es susceptible de aparearse con la cadena (3.104) en virtud del cabo $C_{23} C_{24} C_{25} C_{26}$; a saber, la molécula de detección D_3 (3.70). Una vez que esto sucede, la cadena se ve así:

$$\begin{array}{|c|c|c|} \hline y & \begin{array}{|c|c|c|} \hline \widetilde{C_3 C_4 C_5 C_6 C_7 C_8} \\ \widetilde{C_3 C_4 C_5 C_6 C_7 C_8} \\ \hline \end{array} & 8_{\lambda_3} & \dots \\ \hline \end{array}$$

Y por tanto, el proceso concluye reportándonos que la máquina se detuvo en el estado S_2 .

CASO 3.2: $\lambda_2 = 0$ Es decir, hasta el momento se ha leído la cadena 10; leyéndose actualmente el carácter 0. Así, la cadena (3.103) en realidad es la cadena:

$$A_3 A_4 A_5 A_6 A_7 A_8 \begin{array}{|c|c|c|} \hline \widetilde{A_7 A_8} & 8_{\lambda_3} & 8_{\lambda_4} & \dots \\ \hline \end{array} \quad (3.105)$$

Aquí, la única molécula que se aparea al cabo de la cadena (3.105) es la molécula de transición T_5 (3.62), que es lo esperado según el diagrama del DFA. Así obtenemos la siguiente cadena:

$$\begin{array}{|c|c|c|} \hline \text{RS} & 2 & \begin{array}{|c|c|c|} \hline \widetilde{A_3 A_4 A_5 A_6 A_7 A_8} \\ \widetilde{A_3 A_4 A_5 A_6 A_7 A_8} \\ \hline \end{array} & 8_{\lambda_3} & \dots \\ \hline \end{array} \quad (3.106)$$

CASO 3.3: $\lambda_2 = 1$ Es decir, hasta el momento se han leído los caracteres 11. Entonces la cadena (3.103) en realidad es la cadena:

$$B_3B_4B_5B_6\overbrace{B_7B_8}^{\widetilde{B_7B_8}} \quad \begin{array}{|c|c|c|} \hline 8_{\lambda_3} & 8_{\lambda_4} & \cdots \\ \hline \end{array} \quad (3.107)$$

Se puede observar que la única molécula capaz de adherirse al cabo $B_3B_4B_5B_6$ es la molécula de transición T_6 (3.63). Así obtenemos la siguiente cadena:

$$\begin{array}{|c|c|c|c|c|} \hline \text{RS} & 3 & \overbrace{B_3B_4B_5B_6B_7B_8}^{\widetilde{B_3B_4B_5B_6B_7B_8}} & 8_{\lambda_3} & \cdots \\ \hline \end{array} \quad (3.108)$$

Aquí damos por terminada la prueba, pues al analizar los dos últimos casos, la prueba nos remitirá a alguno de los otros que ya han sido descritos exhaustivamente en esta demostración.

Si nos detenemos en el **CASO 3.2** vemos que la enzima *FokI* al cortar (3.106) deja la cadena:

$$\lambda_2^3\lambda_3^3\lambda_4^3\lambda_5^3\lambda_6^3\lambda_7^3\lambda_8^3 \quad \begin{array}{|c|c|c|} \hline 8_{\lambda_4} & 8_{\lambda_5} & \cdots \\ \hline \end{array} \quad (3.109)$$

La cadena anterior (3.109) no sólo nos indica que la máquina se encuentra en el estado S_3 con entrada λ_3 , en virtud del cabo $\lambda_2^3\lambda_3^3\lambda_4^3\lambda_5^3$, sino que es análoga a la cadena (3.88) heredando así toda su casuística.

Ahora que si nos detenemos en el **CASO 3.3**, la enzima *FokI* corta la cadena (3.108) dejando lo siguiente:

$$\lambda_1^3\lambda_2^3\lambda_3^3\lambda_4^3\lambda_5^3\lambda_6^3\lambda_7^3\lambda_8^3 \quad \begin{array}{|c|c|c|} \hline 8_{\lambda_4} & 8_{\lambda_5} & \cdots \\ \hline \end{array} \quad (3.110)$$

Y el cabo $\lambda_1^3\lambda_2^3\lambda_3^3\lambda_4^3$ nos indica que el DFA está en el estado S_4 con entrada λ_3 . Asimismo, puede observarse que la cadena (3.110) es similar a la cadena (3.95) heredando así toda su casuística.

Por lo tanto, si la máquina es inicializada con la configuración (3.75), entonces está sujeta a los estados y transiciones descritas arriba y a ningún otro; lo que finaliza esta prueba. □

Ahora consideremos una asignación de los moléculas que componen cada una de las cadenas en los conjuntos de $\langle \Gamma, \Delta, \Sigma, \Phi \rangle$. Comenzamos seleccionando los nucleótidos que componen los símbolos del conjunto Σ , y para ello proponemos los símbolos en la figura (3.4)

Símbolo	0	1	t
Versión molecular con codificación de <estado, símbolo> en el cabo pegajoso.			

Figura 3.4: Símbolos y su Codificación de los Estados

Lo cual define las moléculas de transición mostradas en la figura (3.5) y que se construyen siguiendo el esquema mostrado en esta sección.

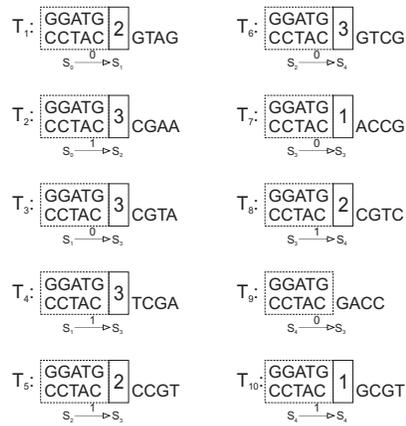


Figura 3.5: Moléculas Transición

Y de igual manera obtenemos las moléculas de detección que se muestran en la figura (3.6).

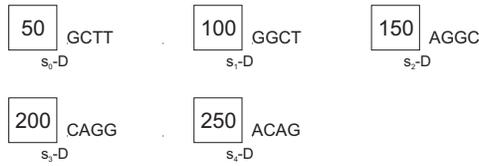


Figura 3.6: Moléculas de Detección

Así, si, por ejemplo, quisiéramos introducir la entrada 110 a esta máquina, tendríamos que construir la molécula de entrada en la forma en que se describe en la figura (3.7).

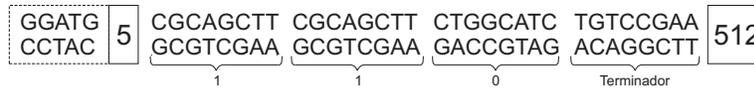


Figura 3.7: Ejemplo de una Molécula de Entrada

De tal forma que al introducir dicha cadena a una solución que contenga a su vez las moléculas de transición, de detección y las enzima de restricción *FokI* y *Ligasa*, esperaríamos que el cómputo procediera en la manera en que se describe en la figura (3.8) al final de esta sección.

Lo anterior completa la descripción del autómata molecular que resuelve la función NAND en su *1a Aproximación*. Como ya se mencionó anteriormente, el trabajo anterior desestima la acción de otras moléculas sobre la máquina (el caso de la *Ligasa* y el *ATP*) y de igual manera estamos trabajando bajo el supuesto de un cómputo serial; en la práctica, en un tubo de ensayo se tienen millones de estas cadenas trabajando en paralelo y por tanto cabe la posibilidad de que muchas de estas cadenas se afecten unas a otras realizando erróneamente el cómputo molecular. Esto es interesante desde el punto de vista de que es necesario un análisis de la tolerancia de errores en una máquina de ADN. Por otro lado, en esta sección se evitó la declaración explícita de las cadenas de ADN que se usaron en el autómata; así, para que este autómata sea implementado en laboratorio será necesario hacer una elección de las cadenas que conformarán el alfabeto de la máquina, es decir, los símbolos 0, 1 y *t*: las cadenas (3.50), (3.51), (3.52), respectivamente. Una vez que se hace una selección de las bases que integrarán estos símbolos, las moléculas de transición y de detección quedan completamente determinadas, salvo las bases que integrarán los espacios arbitrarios en dichas cadenas.

El siguiente capítulo está dedicado a la resolución de esta función en su *2a Aproximación*. Así como para $NAND_1$ se empleó un autómata determinista finito, se usará un autómata mucho más poderoso en términos computacionales; a saber, una *Máquina de Turing*.

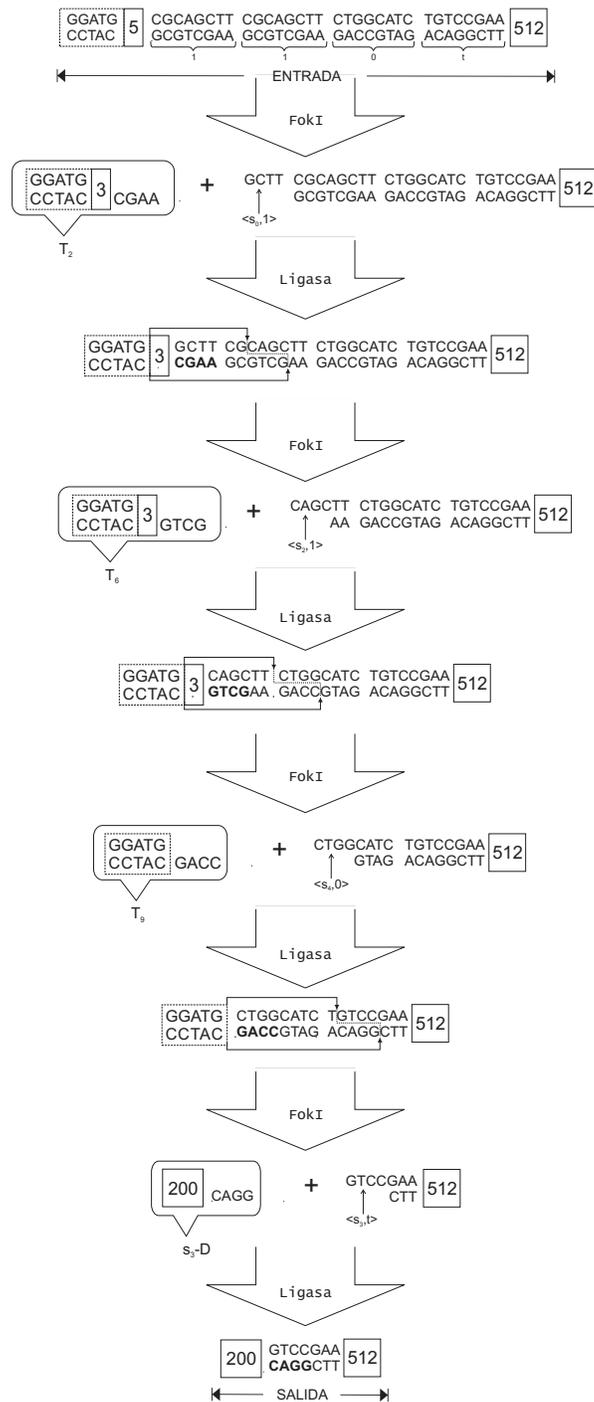


Figura 3.8: Corrida

Capítulo 4

2a Aproximación

Una vez que hemos expuesto la forma en que una máquina sencilla es implementada en moléculas de ADN, podemos considerar la creación de una máquina más compleja. Si para la máquina anterior requerimos únicamente una enzima de restricción que corta cadenas de ADN, en este nuevo caso utilizaremos otras 5 enzimas más, las cuales cortan siguiendo reglas distintas a la que conocimos en el capítulo pasado. De igual forma, permitiremos cortes en ambas direcciones e incrementaremos el número de símbolos de nuestro alfabeto; todo lo anterior para conseguir simular la acción de una *Máquina de Turing* (ver abajo), que a su vez, simula el comportamiento de la función N-AND en su segunda versión. Ésta segunda versión consiste simplemente en tomar dos cadenas binarias de igual longitud y calcular N-AND entrada por entrada. Recordemos que la forma de calcular N-AND es como ya se ha venido describiendo en este trabajo y que se muestra en la tabla (4.1). De tal forma, que la función $NAND_2$ se describe así:

$$NAND_2 : \{0, 1\}^n \times \{0, 1\}^n \longrightarrow \{0, 1\}^n$$

Cuadro 4.1: Tabla de Verdad de la función NAND.

p	q	p NAND q
0	0	1
0	1	1
1	0	1
1	1	0

Como ya se dijo, la manera en que se simulará esta función es a partir de una *Máquina de Turing* (TM, de aquí en adelante) molecular. Entonces, en este capítulo construiremos una TM usando moléculas de ADN, cuyo programa (*software*) resuelve la función $NAND_2$. En resumen, una cadena doble de ADN hará las veces de cinta de la TM, donde también una subcadena de la misma simulará la acción de la cabeza de lectura/escritura; y así como en el capítulo pasado, los elementos del alfabeto, las transiciones y los estados estarán codificados molecularmente con el objetivo de que se imite la lectura/escritura y control, a través de las operaciones de

cortar y pegar cadenas. Pero primero empezaremos por describir matemáticamente la TM que resuelve la función N-AND.

4.1. TM *in abstracto*

Una TM modela matemáticamente la operación de una máquina que opera sobre una cinta sobre la cual se leen y escriben símbolos por medio de una *cabeza de lectura/escritura* que obedece un conjunto finito de reglas. En teoría, la cinta es de longitud infinita y la cabeza puede moverse libremente a lo largo de ella siguiendo el programa determinado por el conjunto de reglas. En comparación con la máquina del capítulo anterior, el DFA, la máquina de Turing posee mayor poder de cómputo ya que al incluir una cinta en su diseño, provee un mecanismo de almacenamiento de datos.

Definición 4.1.1. Una **Máquina de Turing (TM)** es la séptupla $(S, \Sigma, \beta, \Sigma^*, T, s_0, A)$ que consiste en:

- Un conjunto finito de estados (S).
- Un conjunto finito de símbolos llamada alfabeto (Σ).
- Un símbolo $\beta \in \Sigma$ que denota el carácter vacío.
- Un conjunto de símbolos de entrada ($\Sigma^* \subset \Sigma \setminus \{\beta\}$).
- Una función de transición ($T : S \times \Sigma \rightarrow S \times \Sigma \times \{d, i\}$, donde d denota el movimiento a la derecha de la cabeza de lectura/escritura e i a la izquierda).
- Un estado inicial ($s_0 \in S$).
- Un conjunto finito de estados de aceptación ($A \subset S$). A diferencia del DFA, los estados de aceptación en una TM detienen el funcionamiento de la máquina, es decir, cuando se alcanza uno de estos estados, la TM deja de leer/escribir sobre la cinta, por eso a estos estados se les conoce también como estados de detención o paro.

Ahora pasemos a considerar la aplicación de una TM a la simulación de la compuerta N-AND por medio de ADN. Como se ha venido haciendo a lo largo de este trabajo, se tomarán ciertas convenciones para conseguir dicha simulación. La primera de ellas es la manera de escribir las cadenas sobre la cinta de la TM, la cual se describe a continuación.

Sean $a = (a_1, \dots, a_n)$, $b = (b_1, \dots, b_n)$ dos cadenas donde $a_i, b_i \in \{0, 1\}$. Sabemos que

$$c = aNAND_2b = (a_1NANDb_1, \dots, a_nNANDb_n), \quad (4.1)$$

entonces nosotros escribiremos las cadenas a y b sobre la cinta de la TM de la siguiente manera:

...	a_1	b_1	a_2	b_2	...	a_n	b_n	...
-----	-------	-------	-------	-------	-----	-------	-------	-----

(4.2)

De tal manera, que para nuestra TM consideraremos el siguiente alfabeto:

$$\Sigma = \{0, 1, \varepsilon, \beta\} \quad (4.3)$$

Donde, 0 y 1 son los dos caracteres que forman las cadenas sobre las que se calculará la función $NAND_2$; ε es un símbolo que servirá para detectar errores durante el cómputo; mientras que β es el carácter que denota el *espacio en blanco* sobre la cinta de la máquina. Entonces, tenemos que $\Sigma^* = \{0, 1\}$.

Consideraremos el siguiente conjunto de estados:

$$S = \{S_0, S_1, S_2, HALT\} \quad (4.4)$$

Siendo el estado $HALT \in A$ el que representa el paro de la máquina (equivalente al estado de aceptación de un DFA) y S_0 es el estado inicial. De tal manera que para simular la función $NAND_2$ consideraremos las siguientes transiciones:

$$\langle S_0, 0 \rangle \mapsto \langle S_1, \beta, \Rightarrow \rangle \quad (4.5)$$

$$\langle S_0, 1 \rangle \mapsto \langle S_2, \beta, \Rightarrow \rangle \quad (4.6)$$

$$\langle S_0, \beta \rangle \mapsto \langle HALT \rangle \quad (4.7)$$

$$\langle S_1, 0 \rangle \mapsto \langle S_0, 1, \Rightarrow \rangle \quad (4.8)$$

$$\langle S_1, 1 \rangle \mapsto \langle S_0, 1, \Rightarrow \rangle \quad (4.9)$$

$$\langle S_1, \beta \rangle \mapsto \langle S_0, \varepsilon, \Rightarrow \rangle \quad (4.10)$$

$$\langle S_2, 0 \rangle \mapsto \langle S_0, 1, \Rightarrow \rangle \quad (4.11)$$

$$\langle S_2, 1 \rangle \mapsto \langle S_0, 0, \Rightarrow \rangle \quad (4.12)$$

$$\langle S_2, \beta \rangle \mapsto \langle S_0, \varepsilon, \Rightarrow \rangle \quad (4.13)$$

Debe recordarse que la manera de leer estas transiciones es la siguiente:

$$\langle \text{estadoActual}, \text{car'acterEntrante} \rangle \mapsto \langle \text{estadoNuevo}, \text{car'acterNuevo}, \text{moverseHacia} \rangle$$

Así, la transición (4.5) dice que si la máquina está leyendo actualmente el carácter 0 y se encuentra en el estado S_0 , entonces la máquina debe escribir el carácter β en la posición actual, además de cambiarse al estado S_1 y tras hacer esto debe mover la cabeza de lectura/escritura una posición hacia la derecha. Ahora, para la transición (4.7), la interpretación es un poco diferente; para este caso, se debe leer: si la máquina está leyendo actualmente el carácter β y se encuentra en el estado S_0 , entonces la máquina debe detenerse en ese punto, dando por concluido el cómputo.

A continuación se da un ejemplo de cómo funciona la máquina con estas consideraciones.

Sean $a = 1010$ y $b = 1110$ la entrada a $NAND_2$. Por tanto esperamos que la salida sea $c = aNAND_2b = 0101$. Según lo descrito en (4.2), la cinta quedaría inicializada de la siguiente manera:

...	β	1	1	0	1	1	1	0	0	β	...
-----	---------	----------	---	---	---	---	---	---	---	---------	-----

Y la cabeza de lectura/escritura de la TM se encuentra en el estado S_0 y *sobre* el primer carácter distinto de β y de ε leyendo de izquierda a derecha (en este ejemplo, la cabeza está encima del carácter 1 en **negritas**). Así, siguiendo las transiciones anteriores, el cómputo se efectúa de la siguiente manera:

1. Se lee el carácter 1 en el estado S_0 . Aplicando la transición (4.6), la máquina cambia al estado S_2 , escribe β en la posición actual y se mueve una posición a la derecha. La cadena queda:

...	β	β	1	0	1	1	1	0	0	β	...
-----	---------	---------	----------	---	---	---	---	---	---	---------	-----

(Recuérdese que la cabeza de lectura/escritura se encuentra sobre el carácter en **negritas**.)

2. Se lee el carácter 1 en el estado S_2 , en virtud de la transición (4.12) la cadena queda:

...	β	β	0	0	1	1	1	0	0	β	...
-----	---------	---------	---	----------	---	---	---	---	---	---------	-----

3. Se lee el carácter 0 en el estado S_0 , en virtud de la transición (4.5) la cadena queda:

...	β	β	0	β	1	1	1	0	0	β	...
-----	---------	---------	---	---------	----------	---	---	---	---	---------	-----

4. Se lee el carácter 1 en el estado S_1 , en virtud de la transición (4.9) la cadena queda:

...	β	β	0	β	1	1	1	0	0	β	...
-----	---------	---------	---	---------	---	----------	---	---	---	---------	-----

5. Se lee el carácter 1 en el estado S_0 , en virtud de la transición (4.6) la cadena queda:

...	β	β	0	β	1	β	1	0	0	β	...
-----	---------	---------	---	---------	---	---------	----------	---	---	---------	-----

6. Se lee el carácter 1 en el estado S_2 , en virtud de la transición (4.12) la cadena queda:

...	β	β	0	β	1	β	0	0	0	β	...
-----	---------	---------	---	---------	---	---------	---	----------	---	---------	-----

7. Se lee el carácter 0 en el estado S_0 , en virtud de la transición (4.5) la cadena queda:

...	β	β	0	β	1	β	0	β	0	β	...
-----	---------	---------	---	---------	---	---------	---	---------	----------	---------	-----

8. Se lee el carácter 0 en el estado S_1 , en virtud de la transición (4.8) la cadena queda:

...	β	β	0	β	1	β	0	β	1	β	...
-----	---------	---------	---	---------	---	---------	---	---------	---	---------	-----

Y actualmente se está leyendo el carácter β con estado S_0 .

9. Se lee el carácter β en el estado S_0 , en virtud de la transición (4.7) la TM se detiene. En la cinta se puede leer lo siguiente: $\beta\beta 0\beta 1\beta 0\beta 1\beta$; desestimando el carácter en blanco β , la respuesta de la TM es 0101 que es lo que se esperaba.

Ahora mostremos cómo actúa el carácter ε que, como se comentó, sirve para detectar errores durante el cómputo. Sea $a = 1010$ y $b = 1$. Como puede verse, estas cadenas tienen diferente longitud y por ende no es posible realizar su cálculo $NAND_2$; esto será detectado también por nuestra TM. Según lo descrito en (4.2), la cinta quedaría esta vez inicializada de la siguiente manera:

...	β	1	1	0	1	0	β	...
-----	---------	----------	----------	----------	----------	----------	---------	-----

Las transiciones se llevan a cabo de la siguiente forma:

1. Se lee el carácter **1** en el estado S_0 . Aplicando la transición (4.6) la cadena queda:

...	β	β	1	0	1	0	β	...
-----	---------	---------	----------	----------	----------	----------	---------	-----

2. Se lee el carácter **1** en el estado S_2 . Aplicando la transición (4.12) la cadena queda:

...	β	β	0	0	1	0	β	...
-----	---------	---------	----------	----------	----------	----------	---------	-----

3. Se lee el carácter **0** en el estado S_0 . Aplicando la transición (4.5) la cadena queda:

...	β	β	0	β	1	0	β	...
-----	---------	---------	----------	---------	----------	----------	---------	-----

4. Se lee el carácter **1** en el estado S_1 . Aplicando la transición (4.9) la cadena queda:

...	β	β	0	β	1	0	β	...
-----	---------	---------	----------	---------	----------	----------	---------	-----

5. Se lee el carácter **0** en el estado S_0 . Aplicando la transición (4.5) la cadena queda:

...	β	β	0	β	1	β	β	...
-----	---------	---------	----------	---------	----------	---------	---------	-----

Y actualmente se está leyendo el carácter β .

6. Se lee el carácter β en el estado S_1 . Aplicando la transición (4.10) la cadena queda:

...	β	β	0	β	1	β	ε	β	...
-----	---------	---------	----------	---------	----------	---------	---------------	---------	-----

Y nuevamente se lee un carácter β

7. Se lee el carácter β en el estado S_0 y por tanto la máquina se detiene.

Al final de este proceso podemos leer (desestimando los espacios en blanco β) sobre la cinta la cadena 01ε . El contenido del carácter ε en la cadena anterior nos reporta que hubo un error durante el proceso.

Ahora que hemos descrito la TM de forma abstracta estamos en condiciones de exponer su implementación molecular. En ésta usaremos estrategias que ya se han empleado en el capítulo anterior.

4.2. TM Molecular

Para implementar la máquina de la sección anterior mediante moléculas de ADN será necesario, como lo fue en la *1a Aproximación*, diseñar moléculas que permitan simular las transiciones de la máquina. Además, será necesario diseñar cada uno de los símbolos del alfabeto Σ y, tomando en cuenta las enzimas de restricción disponibles en la actualidad, diseñar el *hardware*.

Para lograr nuestro objetivo, incorporaremos ideas de la máquina de Rothmund [5]. Para resolver la función $NAND_2$ mediante ADN que simule la máquina descrita arriba, tendremos que utilizar otras enzimas de restricción además de la ya conocida *FokI*, que bien bastó para resolver la *1a Aproximación*. A continuación se muestran estas enzimas de manera abreviada: escribiremos explícitamente el sitio de reconocimiento de cada uno e implícitamente el número de bases que toma en cuenta antes de partir una cadena doble de ADN. Éstas, además de cortar como lo hemos venido haciendo (de izquierda a derecha), cortarán también en dirección contraria (de derecha a izquierda); naturalmente, sólo cuando la zona de reconocimiento (RS) se encuentre *de cabeza*. Así, la siguiente tabla expone ambas situaciones: cuando corta en una dirección y cuando lo hace en otra.

FokI	{	GGATG	NNNNNNNNN
		CCTAC	NNNNNNNNNNNNNN
		NNNNNNNNNNNNNN	CATCC
		NNNNNNNNN	GTAGG
BbvI	{	GCAGC	NNNNNNNNN
		CGTCG	NNNNNNNNNNNNNN
		NNNNNNNNNNNNNN	GCTGC
		NNNNNNNNN	CGACG
BseRI	{	GAGGAG	NNNNNNNNNN
		CTCCTC	NNNNNNNNN
		NNNNNNNNN	CTCCTC
		NNNNNNNNNN	GAGGAG
BsrDI	{	GCAATG	NN
		CGTTAC	—
		—	CATTGC
		NN	GTAACG
BpmI	{	CTGGAG	NNNNNNNNNNNNNNNNNN
		GACCTC	NNNNNNNNNNNNNNNN
		NNNNNNNNNNNNNNNN	CTCCAG
		NNNNNNNNNNNNNNNN	GAGGTC

Todas estas enzimas conforman el conjunto Φ . Para simplificar aún más esta exposición, adoptaremos la siguiente convención con respecto a la manera de representar las cadenas de ADN y, en especial, los sitios de reconocimiento de las enzimas descritas anteriormente.

Notación 4.2.1. Una cadena de ADN será representada como una cadena de caracteres delimitados por corchetes:

$$[\quad]$$

cuando se trate de cadenas de ADN cuyos extremos NO están enlazados, en cuya situación (el extremo izquierdo de la cadena se encuentra unido al extremo derecho) la cadena se escribirá entre paréntesis:

$$(\quad)$$

En la literatura, las cadenas de ADN que se encuentran unidas por los extremos se conocen como *plásmidos*.

Notación 4.2.2. Consideremos la cadena siguiente:

$$\left[\overrightarrow{RS} \right]$$

\overrightarrow{RS} denota el sitio de reconocimiento de la enzima de restricción $RS \in \Phi$ y que cortará de izquierda a derecha. Análogamente, la siguiente cadena:

$$\left[\overleftarrow{RS} \right]$$

denota la acción de la misma enzima RS , pero en esta ocasión, el corte se hará de derecha a izquierda.

Por ejemplo, la cadena $\left[\overrightarrow{FokI} \right]$ representa el sitio de reconocimiento de la enzima $FokI$ cuya acción es de izquierda a derecha; mientras que, la cadena $\left[\overleftarrow{BpmI} \right]$ denota el sitio de reconocimiento de la enzima $BpmI$ actuando de derecha a izquierda.

Ahora, además de aumentar nuestro *conjunto de herramientas*, utilizaremos una longitud de palabra distinta a la usada en la *1a Aproximación*. Para nuestra TM utilizaremos una longitud de palabra de seis pares para cada uno de los símbolos en Σ ; asimismo, incluiremos un *sufijo* (o *cola*) a cada uno de nuestros símbolos implementados en ADN, el cual tendrá la función de indicar el final del carácter. Este indicador será igual para todos los elementos de Σ y constará de cuatro pares de bases.

Tal y como se hizo en el capítulo anterior (3), no enunciaremos explícitamente las bases que integran cada uno de los símbolos, sino que los describiremos abstractamente. Así, los símbolos en Σ se implementan molecularmente de la siguiente manera y tomando en cuenta el sufijo de terminación:

$$0 = \left[\frac{A_1 A_2 A_3 A_4 A_5 A_6}{\widetilde{A}_1 \widetilde{A}_2 \widetilde{A}_3 \widetilde{A}_4 \widetilde{A}_5 \widetilde{A}_6} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F}_1 \widetilde{F}_2 \widetilde{F}_3 \widetilde{F}_4} \right]$$

$$1 = \left[\frac{B_1 B_2 B_3 B_4 B_5 B_6}{\widetilde{B}_1 \widetilde{B}_2 \widetilde{B}_3 \widetilde{B}_4 \widetilde{B}_5 \widetilde{B}_6} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F}_1 \widetilde{F}_2 \widetilde{F}_3 \widetilde{F}_4} \right]$$

$$\beta = \left[\frac{\beta_1 \beta_2 \beta_3 \beta_4 \beta_5 \beta_6}{\widetilde{\beta_1 \beta_2 \beta_3 \beta_4 \beta_5 \beta_6}} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F_1 F_2 F_3 F_4}} \right]$$

$$\varepsilon = \left[\frac{\varepsilon_1 \varepsilon_2 \varepsilon_3 \varepsilon_4 \varepsilon_5 \varepsilon_6}{\widetilde{\varepsilon_1 \varepsilon_2 \varepsilon_3 \varepsilon_4 \varepsilon_5 \varepsilon_6}} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F_1 F_2 F_3 F_4}} \right]$$

También basándonos en el funcionamiento del DFA de la *1a Aproximación*, el estado actual de la máquina quedará descrito por el cabo pegajoso de la molécula que ha sido partida por alguna de las enzimas en Φ . Esto se consigue de la siguiente manera. Consideremos la siguiente cadena, la cual puede ser cualquiera del conjunto Σ :

$$N = \left[\frac{N_1 N_2 N_3 N_4 N_5 N_6}{\widetilde{N_1 N_2 N_3 N_4 N_5 N_6}} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F_1 F_2 F_3 F_4}} \right]$$

1. La siguiente situación reporta que la máquina se encuentra en el estado S_0 con entrada N .

$$N = \left[\frac{N_1 N_2 N_3 N_4 N_5 N_6}{\widetilde{N_5 N_6}} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F_1 F_2 F_3 F_4}} \right]$$

2. La siguiente situación reporta que la máquina se encuentra en el estado S_1 con entrada N .

$$N = \left[\frac{N_2 N_3 N_4 N_5 N_6}{\widetilde{N_6}} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F_1 F_2 F_3 F_4}} \right]$$

3. La siguiente situación reporta que la máquina se encuentra en el estado S_2 con entrada N .

$$N = \left[\frac{N_3 N_4 N_5 N_6}{\widetilde{N_6}} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F_1 F_2 F_3 F_4}} \right]$$

Recordemos que para la demostración de la afirmación (3.2.1) en el capítulo anterior, fue necesario usar bloques como el siguiente:

$$\boxed{8_{\lambda_i}}$$

Ahora, haremos algo similar para representar las bases que integran cada uno de los símbolos en Σ . Hecho esto, la exhibición de las moléculas de transición se vuelve una tarea más sencilla.

Notación 4.2.3. *La cadena*

$$[X_Y]$$

con $X \in \mathbb{N}$, $Y \in \Sigma$ denota X pares de moléculas de ADN correspondientes al símbolo Y colocados en esa porción de la cadena. Mientras que la cadena

$$[X]$$

solamente indica que en esa posición están colocados X pares arbitrarios.

Por ejemplo, la cadena

$$[6_1|4_F]$$

es la forma simplificada de la cadena de ADN:

$$\left[\frac{A_1 A_2 A_3 A_4 A_5 A_6}{\widetilde{A}_1 \widetilde{A}_2 \widetilde{A}_3 \widetilde{A}_4 \widetilde{A}_5 \widetilde{A}_6} \mid \frac{F_1 F_2 F_3 F_4}{\widetilde{F}_1 \widetilde{F}_2 \widetilde{F}_3 \widetilde{F}_4} \right]$$

Y la cadena

$$[6]$$

es la forma simplificada de la cadena de ADN:

$$\left[\frac{N_1 N_2 N_3 N_4 N_5 N_6}{\widetilde{N}_1 \widetilde{N}_2 \widetilde{N}_3 \widetilde{N}_4 \widetilde{N}_5 \widetilde{N}_6} \right]$$

con $N_i \in \{A, C, G, T\}$ y \widetilde{N}_i es la base complementaria de N_i , para toda $i \in \{1, \dots, 6\}$.

Habiendo expuesto la notación anterior, las moléculas de transición que conforman el conjunto Γ quedan como sigue:

$$T_1 : \langle S_0, 0 \rangle \mapsto \langle S_1, \beta, \Rightarrow \rangle$$

$$\left[\overrightarrow{BsrD\dot{I}}|4_F|6_\beta|4_F|6| \overleftarrow{BserI}| \overrightarrow{Fok\dot{I}}|4|4_F|12| \overleftarrow{BpmI}| \overrightarrow{Bpm\dot{I}}|8|6_0|6| \overleftarrow{BbvI} \right] \quad (4.14)$$

$$T_2 : \langle S_0, 1 \rangle \mapsto \langle S_2, \beta, \Rightarrow \rangle$$

$$\left[\overrightarrow{BsrD\dot{I}}|4_F|6_\beta|4_F|6| \overleftarrow{BserI}| \overrightarrow{Fok\dot{I}}|3|4_F|12| \overleftarrow{BpmI}| \overrightarrow{Bpm\dot{I}}|8|6_1|6| \overleftarrow{BbvI} \right] \quad (4.15)$$

$$T_3 : \langle S_0, \beta \rangle \mapsto \langle HALT \rangle$$

$$\left[\overrightarrow{BsrD\dot{I}}|4_F| H A L T |6_\beta|6| \overleftarrow{BbvI} \right] \quad (4.16)$$

$$T_4 : \langle S_1, 0 \rangle \mapsto \langle S_0, 1, \Rightarrow \rangle$$

$$\left[\overrightarrow{BsrD\dot{I}}|4_F|6_1|4_F|6| \overleftarrow{BserI}| \overrightarrow{Fok\dot{I}}|5|4_F|12| \overleftarrow{BpmI}| \overrightarrow{Bpm\dot{I}}|8|6_0|7| \overleftarrow{BbvI} \right] \quad (4.17)$$

$$T_5 : \langle S_1, 1 \rangle \mapsto \langle S_0, 1, \Rightarrow \rangle$$

$$\left[\overrightarrow{BsrD\dot{I}}|4_F|6_1|4_F|6| \overleftarrow{BserI}| \overrightarrow{Fok\dot{I}}|5|4_F|12| \overleftarrow{BpmI}| \overrightarrow{Bpm\dot{I}}|8|6_1|7| \overleftarrow{BbvI} \right] \quad (4.18)$$

$$T_6 : \langle S_1, \beta \rangle \mapsto \langle S_0, \varepsilon, \Rightarrow \rangle$$

$$\left[\overrightarrow{BsrD\dot{I}}|4_F|6_\varepsilon|4_F|6| \overleftarrow{BserI}| \overrightarrow{Fok\dot{I}}|5|4_F|12| \overleftarrow{BpmI}| \overrightarrow{Bpm\dot{I}}|8|6_\beta|7| \overleftarrow{BbvI} \right] \quad (4.19)$$

$$T_7 : \langle S_2, 0 \rangle \mapsto \langle S_0, 1, \Rightarrow \rangle$$

$$\left[\overrightarrow{BsrD\dot{I}}|4_F|6_1|4_F|6| \overleftarrow{BserI}| \overrightarrow{Fok\dot{I}}|5|4_F|12| \overleftarrow{BpmI}| \overrightarrow{Bpm\dot{I}}|8|6_0|8| \overleftarrow{BbvI} \right] \quad (4.20)$$

$$T_8 : \langle S_2, 1 \rangle \mapsto \langle S_0, 0 \rangle \Rightarrow \left[\overrightarrow{BsrD\bar{I}}|4_F|6_1|4_F|6| \overleftarrow{BserI}|\overrightarrow{FokI}|5|4_F|12| \overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_0|8| \overleftarrow{BbvI} \right] \quad (4.21)$$

$$T_9 : \langle S_2, \beta \rangle \mapsto \langle S_0, \varepsilon \rangle \Rightarrow \left[\overrightarrow{BsrD\bar{I}}|4_F|6_\varepsilon|4_F|6| \overleftarrow{BserI}|\overrightarrow{FokI}|5|4_F|12| \overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_\beta|8| \overleftarrow{BbvI} \right] \quad (4.22)$$

De donde la cadena

$$[H A L T]$$

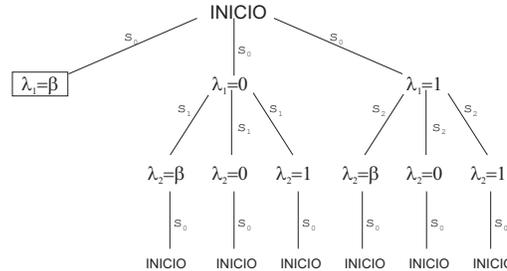
hace las veces de *molécula de detección* para esta máquina. Esta molécula puede diseñarse de la longitud y bases que se deseen con tal de que esta elección no entre en conflicto con alguna otra molécula. La función de ésta es la de reportar que la máquina ha finalizado el cómputo y que lo que se lea en ese momento sobre la cadena de ADN corresponde a la salida del cálculo.

Como pudo observarse en el caso de la *1a Aproximación* y en este caso también, las transiciones se *incluyen* en la alimentación de la máquina. Para el caso de la TM, la cabeza de lectura/escritura también está incluida en cada una de las moléculas de transición, ya que después de la acción de las enzimas de restricción se pierde una parte de la cadena original. Esta cabeza está implementada en las moléculas que denotan el sitio de reconocimiento de las enzimas *FokI* y *BseRI*; la primera corta la entrada de manera tal que permite la descripción del estado actual de la máquina, mientras que la última prepara la entrada para ser enlazada con la siguiente molécula de transición.

Ahora que ya hemos definido cuáles serán las moléculas de transición (el conjunto Γ), las enzimas de restricción (Φ) y el alfabeto de la máquina (Σ) estamos en condiciones de probar que es posible simular molecularmente la función $NAND_2$, imitando el comportamiento de una TM.

Teorema 4.2.1. $\langle \Phi, \Gamma, \Sigma \rangle$ *resuelven la función* $NAND_2$

Demostración. Procederemos como en la demostración de la afirmación (3.2.1) para la generalización de la función $NAND_1$; esto es, suponemos una cadena tan larga como esta prueba lo amerite. En esta prueba seguiremos la ruta trazada por el siguiente árbol.



Tenemos, así, la siguiente configuración inicial:

$$(6_\beta |4_F |6| \overleftarrow{BserI} | \overrightarrow{FokI} |9|6_{\lambda_1} |4_F |6_{\lambda_2} |4_F | \cdots |6_{\lambda_i} |4_F | \cdots) \quad (4.23)$$

Donde decimos que la cabeza de lectura/escritura está posada sobre le carácter λ_1 , pues recordemos que la cabeza está constituida por los sitios de reconocimiento de las enzimas *FokI* (que *apunta* hacia el carácter que está siendo leído actualmente) y *BserI* (que *apunta* al carácter que acaba de ser escrito sobre la cinta). Obsérvese que hemos supuesto que la cadena (4.23) se encuentra unida por sus extremos (ver notación (4.2.1)); a continuación será evidente porque se requiere de esta hipótesis.

El primer paso se realiza cuando las enzimas *FokI* y *BserI* reconocen sus sitios sobre la cadena (4.23) y la cortan como se describió anteriormente (ver arriba) dejando la siguiente cadena:

$$\left(6_\beta | \frac{F_1 F_2}{\quad} | \frac{\lambda_{11} \lambda_{12} \lambda_{13} \lambda_{14} \lambda_{15} \lambda_{16}}{\widetilde{\lambda_{15}} \widetilde{\lambda_{16}}} |4_F |6_{\lambda_2} |4_F | \cdots \right) \quad (4.24)$$

El cabo $\lambda_{11} \lambda_{12} \lambda_{13} \lambda_{14}$ nos indica que la máquina se encuentra en el estado S_0 con entrada λ_1 . Podemos preguntarnos qué ocurriría si la cadena no estuviera unida por sus extremos, es decir, si no se tratara de un *plásmido*. Si ésto no fuera así, después del corte de las enzimas *FokI* y *BserI*, nuestra cadena inicial se partiría en dos cadenas que no podrían ser usadas por estar separadas. Una de estas cadenas sería:

$$\left[6_\beta | \frac{F_1 F_2}{\quad} \right]$$

Y la otra:

$$\left[\frac{\lambda_{11} \lambda_{12} \lambda_{13} \lambda_{14} \lambda_{15} \lambda_{16}}{\widetilde{\lambda_{15}} \widetilde{\lambda_{16}}} |4_F |6_{\lambda_2} |4_F | \cdots \right]$$

dejándonos sin la posibilidad de usarlas para continuar el cómputo. Pero al suponer que la cadena inicial se haya enlazada por los extremos, la acción de ambas enzimas tan sólo deja un *hueco* que se puede apreciar en la cadena (4.24) en la subcadena:

$$\left(\frac{F_1 F_2}{\quad} | \frac{\lambda_{11} \lambda_{12} \lambda_{13} \lambda_{14} \lambda_{15} \lambda_{16}}{\widetilde{\lambda_{15}} \widetilde{\lambda_{16}}} \right)$$

y es donde cabe la posibilidad de proseguir con el cómputo, pues allí puede *cab* alguna de las moléculas del conjunto Γ .

Ahora analicemos la casuística del carácter λ_1 :

CASO 1: $\langle S_0, \lambda_1 = \beta \rangle$ Entonces la cadena (4.24) es en realidad la cadena:

$$\left(6_\beta | \frac{F_1 F_2}{\quad} | \frac{\beta_1 \beta_2 \beta_3 \beta_4 \beta_5 \beta_6}{\widetilde{\beta_5} \widetilde{\beta_6}} |4_F |6_{\lambda_2} |4_F | \cdots \right) \quad (4.25)$$

Entonces esperaríamos que la transición T_3 tuviera lugar. La molécula que representa esta transición se *acomoda* en el *hueco* de la cadena (4.25) como se muestra a continuación:

$$\left(6_\beta | \frac{F_1 F_2}{\quad} | \frac{\beta_1 \beta_2 \beta_3 \beta_4 \beta_5 \beta_6}{\widetilde{\beta_5} \widetilde{\beta_6}} |4_F | \cdots \right) \quad (4.26)$$

$$|\overrightarrow{BsrDI}|4_F | H A L T |6_\beta |6| \overleftarrow{BbvI}|$$

En esta cadena puede identificarse la molécula de transición T_3 (4.16) ocupando el *hueco* en la cadena (4.25), pero sin acoplarse completamente a esta cadena todavía; dicho en otras palabras, sin cubrir los *huecos* en esa cadena; para que eso suceda necesitan actuar las enzimas *BsrDI* y *BbvI*, dejando como resultado la cadena:

$$\left(6_\beta \left| \frac{F_1 F_2}{\quad} \right| \quad \left| \frac{\beta_1 \beta_2 \beta_3 \beta_4 \beta_5 \beta_6}{\widetilde{\beta}_5 \widetilde{\beta}_6} \right| 4_F | \dots \right) \quad (4.27)$$

$$\left| \frac{F_3 F_4}{F_1 \widetilde{F}_2 \widetilde{F}_3 \widetilde{F}_4} \right| H A L T \left| \frac{\quad}{\widetilde{\beta}_1 \widetilde{\beta}_2 \widetilde{\beta}_3 \widetilde{\beta}_4} \right|$$

donde puede verse que la subcadena

$$\left| \frac{F_1 F_2}{\quad} \right|$$

y la subcadena

$$\left| \frac{F_3 F_4}{F_1 \widetilde{F}_2 \widetilde{F}_3 \widetilde{F}_4} \right|$$

son susceptibles de acoplarse en virtud de su cabo pegajoso. Lo mismo puede decirse de la subcadena

$$\left| \frac{\quad}{\widetilde{\beta}_1 \widetilde{\beta}_2 \widetilde{\beta}_3 \widetilde{\beta}_4} \right|$$

y la subcadena

$$\left| \frac{\beta_1 \beta_2 \beta_3 \beta_4 \beta_5 \beta_6}{\widetilde{\beta}_5 \widetilde{\beta}_6} \right|$$

de tal forma que al unirse las subcadenas anteriores, la cadena (4.27) queda:

$$(6_\beta | 4_F | H A L T | 6_\beta | 4_F | \dots) \quad (4.28)$$

resultando en una cadena *sin hoyos*. En este punto el cómputo se detiene pues la cadena (4.28) no es susceptible de ser cortada por alguna de las enzimas en Φ , lo cual concuerda con el funcionamiento esperado de la TM.

CASO 2: $\langle S_0, \lambda_1 = 0 \rangle$ Entonces la cadena (4.24) es en realidad la cadena:

$$\left(6_\beta \left| \frac{F_1 F_2}{\quad} \right| \left| \frac{A_1 A_2 A_3 A_4 A_5 A_6}{\widetilde{A}_5 \widetilde{A}_6} \right| 4_F | 6_{\lambda_2} | 4_F | \dots \right) \quad (4.29)$$

En este caso, la transición T_1 debe ocurrir. Esta transición está representada por la molécula (4.14) y al querer unirse a la cadena (4.29) resulta en la siguiente cadena:

$$\left(6_\beta \left| \frac{F_1 F_2}{\quad} \right| \quad \left| \frac{A_1 A_2 A_3 A_4 A_5 A_6}{\widetilde{A}_5 \widetilde{A}_6} \right| \dots \right)$$

$$\left| \overrightarrow{BsrDI} \right| 4_F | 6_\beta | 4_F | 6 | \overleftarrow{BsrI} | \overrightarrow{FokI} | 4 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | 6_0 | 6 | \overleftarrow{BbvI} |$$

$$(4.30)$$

Al reconocer y cortar las enzimas *BsrDI* y *BbvI* dejan la siguiente cadena:

$$\left(6_\beta \left| \frac{F_1 F_2}{\quad} \right| \quad \left| \frac{A_1 A_2 A_3 A_4 A_5 A_6}{\widetilde{A}_5 \widetilde{A}_6} \right| \dots \right)$$

$$\left| \frac{F_3 F_4}{F_1 \widetilde{F}_2 \widetilde{F}_3 \widetilde{F}_4} \right| 6_\beta | 4_F | 6 | \overleftarrow{BsrI} | \overrightarrow{FokI} | 4 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | \frac{\quad}{\widetilde{A}_1 \widetilde{A}_2 \widetilde{A}_3 \widetilde{A}_4} |$$

$$(4.31)$$

Los cabos de la cadena anterior son susceptibles de aparearse; al hacerlo dejan la siguiente cadena:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|4|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_0|4_F|6_{\lambda_2}|4_F|\dots) \quad (4.32)$$

Ahora, cuando la enzima de destrucción $BpmI$ reconoce ambos sitios en la cadena (4.32) corta esta cadena dejando la siguiente:

$$\left(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|4|\frac{F_1 F_2}{\overline{F_1 F_2 F_3 F_4}}|\frac{F_3 F_4}{\overline{F_1 F_2 F_3 F_4}}|6_{\lambda_2}|4_F|\dots\right) \quad (4.33)$$

cuyos cabos son susceptibles de aparearse resultando en la cadena:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|4|4_F|6_{\lambda_2}|4_F|\dots) \quad (4.34)$$

la cual consideraremos una nueva *configuración inicial* (como en (4.23)) y cuyo desarrollo estudiaremos más adelante en esta prueba. Por el momento diremos que la cabeza se encuentra actualmente sobre el carácter λ_2

CASO 3: $\langle S_0, \lambda_1 = 1 \rangle$ Entonces la cadena (4.24) es en realidad la cadena:

$$\left(6_\beta|\frac{F_1 F_2}{\overline{F_1 F_2 F_3 F_4}}|\frac{B_1 B_2 B_3 B_4 B_5 B_6}{\overline{B_5 B_6}}|4_F|6_{\lambda_2}|4_F|\dots\right) \quad (4.35)$$

En este punto esperamos que la transición T_2 tenga lugar. Veamos si la molécula de transición T_2 (4.15) se *pega* a la cadena anterior. La molécula (4.15) se coloca en el *agujero* de la cadena (4.35) como se muestra en la siguiente cadena:

$$\left(6_\beta|\frac{F_1 F_2}{\overline{F_1 F_2 F_3 F_4}}|\frac{B_1 B_2 B_3 B_4 B_5 B_6}{\overline{B_5 B_6}}|\dots\right) \\ |\overrightarrow{BsrDI}|4_F|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|3|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_1|6|\overleftarrow{BbvI}| \quad (4.36)$$

Una vez que las enzimas $BsrDI$ y $BbvI$ reconocen su sitio en la cadena anterior, la cortan resultando en la cadena:

$$\left(6_\beta|\frac{F_1 F_2}{\overline{F_1 F_2 F_3 F_4}}|\frac{B_1 B_2 B_3 B_4 B_5 B_6}{\overline{B_5 B_6}}|\dots\right) \\ |\frac{F_3 F_4}{\overline{F_1 F_2 F_3 F_4}}|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|3|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|\frac{F_3 F_4}{\overline{B_1 B_2 B_3 B_4}}| \quad (4.37)$$

Los cabos de la cadena anterior son susceptibles de enlazarse lo cual ocurre resultando en la cadena:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|3|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_1|4_F|6_{\lambda_2}|4_F|\dots) \quad (4.38)$$

En esta cadena hay dos sitios de reconocimiento de la enzima de restricción $BpmI$ y cuando son reconocidos por esta enzima, la cadena (4.38) queda:

$$\left(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|3|\frac{F_1 F_2}{\overline{F_1 F_2 F_3 F_4}}|\frac{F_3 F_4}{\overline{F_1 F_2 F_3 F_4}}|6_{\lambda_2}|4_F|\dots\right) \quad (4.39)$$

cuyos cabos son capaces de aparearse y al hacerlo la cadena anterior queda como:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|3|4_F|6_{\lambda_2}|4_F|\dots) \quad (4.40)$$

que, análogamente al **CASO 2**, se trata de una nueva configuración inicial cuyo desarrollo se expondrá más adelante. Actualmente la cabeza está *sobre* el carácter λ_2 y acaba de *escribir* el carácter β tal y como se esperaba.

Con la exposición anterior hemos agotado todas las posibilidades del carácter λ_1 , es tiempo de que desarrollemos cada uno de los casos anteriores para conocer el comportamiento de esta TM Molecular. Como el **CASO 1** carece de sentido puesto que la máquina se ha detenido, analizaremos los dos casos restantes. Comenzaremos por el **CASO 2**, desarrollando la rama de $\lambda_1 = 0$. Tenemos como nueva configuración inicial la cadena:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|4|4_F|6_{\lambda_2}|4_F|\dots) \quad (4.41)$$

sabemos que la cabeza está actualmente posada sobre el carácter λ_2 . Esta cadena es susceptible de ser cortada por las enzimas *FokI* y *BserI*; y cuando esto ocurre, la cadena queda así:

$$\left(6_\beta|4_F|6_\beta|\frac{F_1F_2}{\lambda_{26}}|\frac{\lambda_{22}\lambda_{23}\lambda_{24}\lambda_{25}\lambda_{26}}{\lambda_{26}}|4_F|6_{\lambda_3}|4_F|\dots\right) \quad (4.42)$$

donde el cabo $\lambda_{22}\lambda_{23}\lambda_{24}\lambda_{25}$ nos indica que la TM se encuentra en el estado S_1 con entrada λ_2 tal y como se esperaba al haber leído el carácter 0 sobre la cinta.

Analicemos la casuística del carácter λ_2 .

CASO 2.1: $\langle S_1, \lambda_2 = \beta \rangle$ Entonces la cadena (4.42) es en realidad la cadena:

$$\left(6_\beta|4_F|6_\beta|\frac{F_1F_2}{\tilde{\beta}_6}|\frac{\beta_2\beta_3\beta_4\beta_5\beta_6}{\tilde{\beta}_6}|4_F|6_{\lambda_3}|4_F|\dots\right) \quad (4.43)$$

Según las transiciones de la TM, cuando la máquina lee β con estado S_1 ocurre la transición T_6 , es decir, la molécula (4.19); la cual se coloca en la cadena (4.43) con *intención* de aparearse, lo cual no ocurrirá hasta que las enzimas de restricción reconozcan sus sitios. Mientras, la cadena queda:

$$\left(6_\beta|4_F|6_\beta|\frac{F_1F_2}{\tilde{\beta}_6}|\frac{\beta_2\beta_3\beta_4\beta_5\beta_6}{\tilde{\beta}_6}|\dots\right) \\ |\overrightarrow{BsrDI}|4_F|6_\varepsilon|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_\beta|7|\overleftarrow{BbvI}| \quad (4.44)$$

Una vez que las enzimas *BsrDI* y *BbvI* reconocen y cortan la cadena anterior, la cadena se transforma en la siguiente cadena:

$$\left(6_\beta|4_F|6_\beta|\frac{F_1F_2}{\tilde{\beta}_6}|\frac{\beta_2\beta_3\beta_4\beta_5\beta_6}{\tilde{\beta}_6}|\dots\right) \\ |\frac{F_3F_4}{\tilde{F}_1\tilde{F}_2\tilde{F}_3\tilde{F}_4}|6_\varepsilon|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|\frac{\beta_1}{\tilde{\beta}_1\tilde{\beta}_2\tilde{\beta}_3\tilde{\beta}_4\tilde{\beta}_5}| \quad (4.45)$$

y como se ve, sus cabos son susceptibles de pegarse. Cuando esto sucede, la cadena queda:

$$(6_\beta|4_F|6_\beta|4_F|6_\varepsilon|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_\beta|4_F|6_{\lambda_3}|4_F|\dots) \quad (4.46)$$

Ahora es el turno de la enzima $BpmI$ de reconocer y cortar la cadena anterior. Así, la resulta la siguiente cadena:

$$\left(6_\beta |4_F| 6_\beta |4_F| 6_\varepsilon |4_F| 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | \overrightarrow{F_1 F_2} | \left| \frac{F_3 F_4}{\overleftarrow{F_1} \overleftarrow{F_2} \overleftarrow{F_3} \overleftarrow{F_4}} \right| 6_{\lambda_3} |4_F| \dots \right) \quad (4.47)$$

donde una vez más se puede apreciar la complementariedad de los cabos pegajosos. Cuando éstos se adhieren resulta la siguiente cadena:

$$(6_\beta |4_F| 6_\beta |4_F| 6_\varepsilon |4_F| 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 |4_F| 6_{\lambda_3} |4_F| \dots) \quad (4.48)$$

la cual nuevamente se puede considerar como si fuera una nueva configuración inicial. Aun más, sabemos que la cabeza de la TM está sobre el carácter λ_3 con estado S_0 y que se acaba de escribir el carácter ε sobre la *cinta*. Dejemos por el momento el desarrollo de esta nueva configuración inicial y pongamos atención en los siguientes casos.

CASO 2.2: $\langle S_1, \lambda_2 = 0 \rangle$ Entonces la cadena (4.42) es en realidad la cadena:

$$\left(6_\beta |4_F| 6_\beta | \overrightarrow{F_1 F_2} | \left| \frac{A_2 A_3 A_4 A_5 A_6}{\overleftarrow{A_6}} \right| 4_F | 6_{\lambda_3} |4_F| \dots \right) \quad (4.49)$$

Sabemos, por el estado actual de la máquina y el carácter que se lee actualmente, que la transición que aplica es T_4 ; y así, la molécula que entra en juego es la molécula (4.17). Así, la nueva cadena se ve:

$$\left(6_\beta |4_F| 6_\beta | \overrightarrow{F_1 F_2} | \left| \frac{A_2 A_3 A_4 A_5 A_6}{\overleftarrow{A_6}} \right| \dots \right) \\ | \overrightarrow{BsrDI} | 4_F | 6_1 | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | 6_0 | 7 | \overleftarrow{BbvI} | \quad (4.50)$$

Primero, sobre esta cadena, actúan las enzimas $BsrDI$ y $BbvI$ resultando en la cadena:

$$\left(6_\beta |4_F| 6_\beta | \overrightarrow{F_1 F_2} | \left| \frac{A_2 A_3 A_4 A_5 A_6}{\overleftarrow{A_6}} \right| \dots \right) \\ \left| \frac{F_3 F_4}{\overleftarrow{F_1} \overleftarrow{F_2} \overleftarrow{F_3} \overleftarrow{F_4}} \right| 6_1 | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | \frac{A_1}{\overleftarrow{A_1} \overleftarrow{A_2} \overleftarrow{A_3} \overleftarrow{A_4} \overleftarrow{A_5}} | \quad (4.51)$$

que tiene dos partes susceptibles de aparearse en virtud de sus cabos. Cuando estos cabos de pegan, la cadena queda:

$$(6_\beta |4_F| 6_\beta |4_F| 6_1 |4_F| 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | 6_0 | 4_F | 6_{\lambda_3} |4_F| \dots) \quad (4.52)$$

El siguiente paso es dejar que la enzima $BpmI$ reconozca y corte en el lugar que deba sobre la cadena anterior. Cuando ésto ocurre la cadena se ve así:

$$\left(6_\beta |4_F| 6_\beta |4_F| 6_1 |4_F| 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | \overrightarrow{F_1 F_2} | \left| \frac{F_3 F_4}{\overleftarrow{F_1} \overleftarrow{F_2} \overleftarrow{F_3} \overleftarrow{F_4}} \right| 6_{\lambda_3} |4_F| \dots \right) \quad (4.53)$$

Una vez más, la acción de las enzimas de restricción dejan la posibilidad de que los cabos se apareen y entonces la cadena resultante es:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|4_F|6_{\lambda_3}|4_F|\dots) \quad (4.54)$$

En la cadena anterior podemos ver que la cabeza está leyendo el carácter λ_3 y recién escribió el carácter 1 que es lo que se espera al haber leído la cadena 00. El nuevo estado es el estado S_0 y más adelante se describirá qué ocurre al desarrollar esta nueva configuración inicial.

CASO 2.3: $\langle S_1, \lambda_2 = 1 \rangle$ Entonces la cadena (4.42) es en realidad la cadena:

$$\left(6_\beta|4_F|6_\beta|\overrightarrow{F_1F_2}| \left| \overrightarrow{\frac{B_2B_3B_4B_5B_6}{\overline{B_6}}} \right| 4_F|6_{\lambda_3}|4_F|\dots \right) \quad (4.55)$$

Con entrada $\lambda_2 = 1$ y estado S_1 la transición esperada es T_5 , es decir, la molécula (4.18). Cuando esta molécula pretende aparearse con la cadena (4.55) la cadena se ve así:

$$\left(6_\beta|4_F|6_\beta|\overrightarrow{F_1F_2}| \left| \overrightarrow{\frac{B_2B_3B_4B_5B_6}{\overline{B_6}}} \right| \dots \right) \\ \left| \overrightarrow{BsrDI} \right| 4_F|6_1|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_1|7|\overleftarrow{BbvI}| \quad (4.56)$$

Como ya se ha venido diciendo, las enzimas $BsrDI$ y $BbvI$ reconocen sus respectivos sitios en la cadena anterior; y al cortar deja la siguiente cadena:

$$\left(6_\beta|4_F|6_\beta|\overrightarrow{F_1F_2}| \left| \overrightarrow{\frac{B_2B_3B_4B_5B_6}{\overline{B_6}}} \right| \dots \right) \\ \left| \overrightarrow{\frac{F_3F_4}{\overline{F_1F_2F_3F_4}}} \right| 6_1|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|\overrightarrow{\frac{B_1}{\overline{B_1B_2B_3B_4B_5}}} \quad (4.57)$$

De tal forma que sus cabos se pegan formando la siguiente cadena:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_1|4_F|6_{\lambda_3}|4_F|\dots) \quad (4.58)$$

Cuando la enzima $BpmI$ reconoce y corta como es debido, la cadena resultante es:

$$\left(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|\overrightarrow{F_1F_2}| \left| \overrightarrow{\frac{F_3F_4}{\overline{F_1F_2F_3F_4}}} \right| 6_{\lambda_3}|4_F|\dots \right) \quad (4.59)$$

Cuyos cabos se pegan nuevamente dejando la siguiente cadena:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{BseRI}|\overrightarrow{FokI}|5|4_F|6_{\lambda_3}|4_F|\dots) \quad (4.60)$$

La cual también puede considerarse una nueva configuración inicial. Y la cabeza se encuentra sobre el carácter λ_3 con estado S_0 tras escribir el carácter 1 en la cinta, tal y como se esperaba después de haber leído 01.

Podemos preguntarnos qué ocurre al desarrollar los tres casos anteriores, pero la respuesta a esta pregunta se encuentra en la siguiente observación.

Observación 4.2.1. *Echemos un vistazo a las cintas que resultan de aplicar los 3 anteriores CASOS (2.1,2.2,2.3). Lo primero que podemos observar es que las*

cadena (4.54) y (4.60) son iguales; y las tres, las cadenas (4.48), (4.54) y (4.60) corresponden al estado S_0 con entrada λ_3 difiriendo únicamente en el contenido de su cinta. Estas cadenas se parecen a la configuración inicial de la máquina, la cadena (4.23) y por lo tanto, podríamos esperar que, no obstante lo que hallemos en la cinta, los resultados serían iguales a los ya expuestos. Es decir, esperaríamos el mismo movimiento molecular que se ha descrito en los casos anteriores. Esto en lo que toca al suponer que $\lambda_1 = 0$. Si suponemos que $\lambda_1 = 1$ podremos agotar todas los posibles caminos que tome la máquina. A continuación se muestra el desarrollo de esta rama.

Tenemos la siguiente cadena como nueva configuración inicial, donde se ve que la cabeza se encuentra sobre el carácter λ_2 .

$$(6_\beta | 4_F | 6_\beta | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 3 | 4_F | 6_{\lambda_2} | 4_F | \dots) \quad (4.61)$$

Sobre la cual las enzimas $FokI$ y $BserI$ reconocen y cortan. La cadena restante es:

$$\left(6_\beta | 4_F | 6_\beta | \overrightarrow{F_1 F_2} | | \overrightarrow{\lambda_{23} \lambda_{24} \lambda_{25} \lambda_{26}} | 4_F | 6_{\lambda_3} | 4_F | \dots \right) \quad (4.62)$$

Donde el cabo $\lambda_{23} \lambda_{24} \lambda_{25} \lambda_{26}$ nos dice que la máquina se encuentra actualmente en el estado S_2 con entrada λ_2

Nuevamente nos preguntamos por el valor del carácter λ_2 y obtenemos la siguiente casuística:

CASO 3.1: $\langle S_2, \lambda_2 = \beta \rangle$ Entonces la cadena (4.62) es en realidad la cadena:

$$\left(6_\beta | 4_F | 6_\beta | \overrightarrow{F_1 F_2} | | \overrightarrow{\beta_3 \beta_4 \beta_5 \beta_6} | 4_F | 6_{\lambda_3} | 4_F | \dots \right) \quad (4.63)$$

Y esperamos que ocurra la transición T_9 dada la entrada $\lambda_2 = \beta$ y el estado S_2 . La molécula que representa esta transición es (4.22) la cual al juntarse con la cadena anterior resulta en la siguiente cadena:

$$\left(6_\beta | 4_F | 6_\beta | \overrightarrow{F_1 F_2} | | \overrightarrow{\beta_3 \beta_4 \beta_5 \beta_6} | \dots \right) \\ | \overrightarrow{BsrDI} | 4_F | 6_\varepsilon | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | 6_\beta | 8 | \overleftarrow{BbvI} | \quad (4.64)$$

Para que esta molécula de transición pueda pegarse adecuadamente a la cadena (4.63) es necesario que las enzimas $FokI$ y $BserI$ actúen sobre ella. Cuando ésto sucede la cadena resultante es:

$$\left(6_\beta | 4_F | 6_\beta | \overrightarrow{F_1 F_2} | | \overrightarrow{\beta_3 \beta_4 \beta_5 \beta_6} | \dots \right) \\ | \overrightarrow{\frac{F_3 F_4}{\tilde{F}_1 \tilde{F}_2 \tilde{F}_3 \tilde{F}_4}} | 6_\varepsilon | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | \overrightarrow{\frac{\beta_1 \beta_2}{\tilde{\beta}_1 \tilde{\beta}_2 \tilde{\beta}_3 \tilde{\beta}_4 \tilde{\beta}_5 \tilde{\beta}_6}} | \quad (4.65)$$

la cual tiene zonas con cabos pegajosos complementarios, y en virtud de esta complementariedad se adhieren unos con otros resultando en la siguiente cadena:

$$(6_\beta | 4_F | 6_\beta | 4_F | 6_\varepsilon | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | 6_\beta | 4_F | 6_{\lambda_3} | 4_F | \dots) \quad (4.66)$$

El siguiente paso es que la enzima $BpmI$ reconozca su sitio correspondiente en la cadena anterior y corte como le corresponde. Cuando esto pasa, la cadena se transforma en la siguiente cadena:

$$\left(6_\beta | 4_F | 6_\beta | 4_F | 6_\varepsilon | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | \overrightarrow{F_1 F_2} | \left| \frac{F_3 F_4}{\widetilde{F_1} \widetilde{F_2} \widetilde{F_3} \widetilde{F_4}} \right| 6_{\lambda_3} | 4_F | \dots \right) \quad (4.67)$$

la cual a su vez presenta lugares con cabos complementarios. Al aparearse estos, la cadena queda:

$$(6_\beta | 4_F | 6_\beta | 4_F | 6_\varepsilon | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 6_{\lambda_3} | 4_F | \dots) \quad (4.68)$$

la cual puede considerarse como una nueva configuración inicial, cuyo desarrollo se describirá más adelante; por lo pronto, sabemos que la cabeza se encuentra encima del carácter λ_3 y acaba de escribir el carácter ε sobre la cinta, tal y como se esperaba por la entrada 1β .

CASO 3.2: $\langle S_2, \lambda_2 = 0 \rangle$ Entonces la cadena (4.62) es en realidad la cadena:

$$\left(6_\beta | 4_F | 6_\beta | \overrightarrow{F_1 F_2} | \left| \frac{A_3 A_4 A_5 A_6}{\dots} \right| 4_F | 6_{\lambda_3} | 4_F | \dots \right) \quad (4.69)$$

Y para la entrada $\lambda_2 = 0$ con estado S_2 esperamos que la transición T_7 tenga lugar. Así la cadena anterior se transforma en la siguiente cadena:

$$\left(6_\beta | 4_F | 6_\beta | \overrightarrow{F_1 F_2} | \left| \frac{A_3 A_4 A_5 A_6}{\dots} \right| \dots \right) \\ \left| \overrightarrow{BsrDI} \right| 4_F | 6_1 | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | 6_0 | 8 | \overleftarrow{BbvI} | \quad (4.70)$$

Sobre la cual actúan las enzimas de restricción $FokI$ y $BserI$ dejando la cadena en la siguiente forma:

$$\left(6_\beta | 4_F | 6_\beta | \overrightarrow{F_1 F_2} | \left| \frac{A_3 A_4 A_5 A_6}{\dots} \right| \dots \right) \\ \left| \frac{F_3 F_4}{\widetilde{F_1} \widetilde{F_2} \widetilde{F_3} \widetilde{F_4}} \right| 6_1 | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | \frac{A_1 A_2}{\widetilde{A_1} \widetilde{A_2} \widetilde{A_3} \widetilde{A_4} \widetilde{A_5} \widetilde{A_6}} | \quad (4.71)$$

que al ser apareados los cabos complementarios resulta en la siguiente cadena:

$$(6_\beta | 4_F | 6_\beta | 4_F | 6_1 | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} | 8 | 6_0 | 4_F | 6_{\lambda_3} | 4_F | \dots) \quad (4.72)$$

Es entonces cuando la enzima $BpmI$ actúa sobre la cadena anterior resultando en la cadena:

$$\left(6_\beta | 4_F | 6_\beta | 4_F | 6_1 | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | \overrightarrow{F_1 F_2} | \left| \frac{F_3 F_4}{\widetilde{F_1} \widetilde{F_2} \widetilde{F_3} \widetilde{F_4}} \right| 6_{\lambda_3} | 4_F | \dots \right) \quad (4.73)$$

Es decir,

$$(6_\beta | 4_F | 6_\beta | 4_F | 6_1 | 4_F | 6 | \overleftarrow{BseRI} | \overrightarrow{FokI} | 5 | 4_F | 6_{\lambda_3} | 4_F | \dots) \quad (4.74)$$

y esta última cadena puede considerarse como una nueva configuración inicial cuyo desarrollo se expondrá más adelante. También puede verse que la cabeza se encuentra actualmente sobre el carácter λ_3 y se acaba de escribir el carácter 1 sobre la cinta.

CASO 3.3: $\langle S_2, \lambda_2 = 1 \rangle$ Entonces la cadena (4.62) es en realidad la cadena:

$$\left(6_\beta |4_F |6_\beta | \frac{F_1 F_2}{\overline{\quad}} | | \frac{B_3 B_4 B_5 B_6}{\overline{\quad}} |4_F |6_{\lambda_3} |4_F | \dots \right) \quad (4.75)$$

Para esta entrada y este estado esperamos que la transición T_8 tenga lugar. Cuando la molécula (4.21) se une a la cadena anterior obtenemos la siguiente cadena:

$$\left(6_\beta |4_F |6_\beta | \frac{F_1 F_2}{\overline{\quad}} | \frac{B_3 B_4 B_5 B_6}{\overline{\quad}} | \dots \right) \\ | \overrightarrow{BsrDI} |4_F |6_0 |4_F |6 | \overleftarrow{BseRI} | \overrightarrow{FokI} |5 |4_F |12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} |8 |6_1 |8 | \overleftarrow{BbvI} | \quad (4.76)$$

El primer paso se da cuando las enzimas $FokI$ y $BserI$ reconocen y cortan sobre esta última cadena. Al hacerlo, la cadena queda así:

$$\left(6_\beta |4_F |6_\beta | \frac{F_1 F_2}{\overline{\quad}} | \frac{B_3 B_4 B_5 B_6}{\overline{\quad}} | \dots \right) \\ | \frac{F_3 F_4}{\overline{F_1 F_2 F_3 F_4}} |6_0 |4_F |6 | \overleftarrow{BseRI} | \overrightarrow{FokI} |5 |4_F |12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} |8 | \frac{B_1 B_2}{\overline{B_1 B_2 B_3 B_4 B_5 B_6}} | \quad (4.77)$$

Cuando los cabos de la cadena anterior se adhieren la cadena queda así:

$$(6_\beta |4_F |6_\beta |4_F |6_0 |4_F |6 | \overleftarrow{BseRI} | \overrightarrow{FokI} |5 |4_F |12 | \overleftarrow{BpmI} | \overrightarrow{BpmI} |8 |6_1 |4_F |6_{\lambda_3} |4_F | \dots) \quad (4.78)$$

Y cuando la enzima $BpmI$ reconoce su sitio sobre la cadena anterior, la cadena resultante es:

$$\left(6_\beta |4_F |6_\beta |4_F |6_0 |4_F |6 | \overleftarrow{BseRI} | \overrightarrow{FokI} |5 | \frac{F_1 F_2}{\overline{\quad}} | | \frac{F_3 F_4}{\overline{F_1 F_2 F_3 F_4}} |6_{\lambda_3} |4_F | \dots \right) \quad (4.79)$$

Al pegarse los cabos complementarios, resulta:

$$(6_\beta |4_F |6_\beta |4_F |6_0 |4_F |6 | \overleftarrow{BseRI} | \overrightarrow{FokI} |5 |4_F |6_{\lambda_3} |4_F | \dots) \quad (4.80)$$

cadena que nuevamente puede considerarse como una nueva configuración inicial. En esta cadena se puede observar que la cabeza está sobre el carácter λ_3 y que se acaba de escribir el carácter 0 sobre la cinta.

Nuevamente, es natural preguntarse qué ocurre cuando desarrollamos los **CASOS** anteriores. La respuesta a esta pregunta se encuentra en la siguiente observación.

Observación 4.2.2. *Echemos un vistazo a las cintas que resultan de aplicar los CASOS 3.1, 3.2 y 3.3, es decir, a las cadenas (4.68), (4.74) y (4.80). Y tal como ocurría en la observación anterior, éstas son muy similares entre sí y a la configuración inicial (4.23). Así, podemos esperar el mismo comportamiento de la máquina para lo que resta de la operación. En otras palabras, que siempre el funcionamiento de la máquina se remitirá a alguno de los casos ya descritos.*

Lo anterior no sólo prueba que las moléculas de transición se aplican correctamente sobre la cinta molecular, sino que éstas escriben lo esperado sobre la cinta. \square

Con la demostración de la afirmación (3.2.1) y el teorema (4.2.1) hemos probado que en cada caso nuestra selección de moléculas de transición, moléculas de detección, alfabeto y enzimas de restricción, consigue simular adecuadamente dos versiones de la función booleana NAND y así mostrar cómo es posible crear una *unidad lógica molecular* que es parte esencial de cualquier dispositivo de cómputo. Debe de quedar claro que no obstante la capacidad de estas moléculas operando en conjunto, el cómputo depende también de la selección que se haga de las moléculas arbitrarias presentes en las cadenas de ADN descritas en ambos casos ($NAND_1$ y $NAND_2$); esta selección se debe hacer cuidando que no entre en conflicto con las moléculas operantes, es decir, las encargadas de realizar el proceso; por ejemplo, se debe evitar colocar en esos sitios las moléculas que son reconocidas por alguna enzima de restricción cuando corta de izquierda a derecha ni viceversa. Es necesario mencionar también que la notación empleada para la exposición de esta TM se puede usar también para la exposición de los autómatas del capítulo anterior (3); y así, a lo largo de esta tesis hemos venido mejorando nuestra notación presentando los pasos intermedios hasta lograr una mejor abstracción de estas cadenas de ADN.

En el capítulo siguiente se describirán las conclusiones de este trabajo, así como una exposición del futuro que posiblemente tenga este nuevo modelo de cómputo además de una discusión de los problemas abiertos en este campo. Por lo pronto, podemos revisar una corrida de esta máquina, pero, debido a la extensión de las moléculas de transición (ver arriba), usaremos una visualización alterna a la asignación de nucleótidos y que nos facilitará esta exposición. Dicha visualización es como sigue: consideraremos cada uno de los elementos del alfabeto como un bloque de líneas verticales, horizontales o inclinadas, como se muestra en la figura (4.1). En esta figura representamos cada uno de los símbolos de entrada ($0, 1, \varepsilon$ y β) recordando que cada uno de ellos consta de 6 nucleótidos, a los cuales se les agrega el sufijo F al final, el cual consta de otros 4 nucleótidos.



Figura 4.1: Símbolos de la Máquina de Turing y su Codificación de las Entradas

Si procedemos de esta manera, también representaremos los sitios de reconocimiento de las enzimas en el conjunto Φ de forma que tanto su composición como el sentido en que realizan su corte queden abreviados; y esto se muestra en la figura (4.2).

REPRESENTACIÓN	SENTIDO DEL CORTE	SITIO DE RECONOCIMIENTO
	izquierda a derecha	GGATG CCTAC
	derecha a izquierda	CTCCTC GAGGAG
	izquierda a derecha	GCAATG CGTTAC
	derecha a izquierda	GCTGC CGACG
	izquierda a derecha	CTGGAG GACCTC
	derecha a izquierda	CTCCAG GAGGTC

Figura 4.2: Enzimas de Restricción

Así, escribimos de forma abreviada cada una de las moléculas de transición como se muestra en la figura (4.4). Un ejemplo de cadena de entrada para esta máquina molecular se muestra en la figura (4.3); en este caso se escribió la cadena 0110, que es la convención de escribir las cadenas $a = 01$ y $b = 10$ sobre la cinta. Hay que recordar que la entrada de esta máquina es un *plásmido*, o sea, una cadena doble de ADN pegada por sus extremos.



Figura 4.3: Ejemplo de una Molécula de Entrada

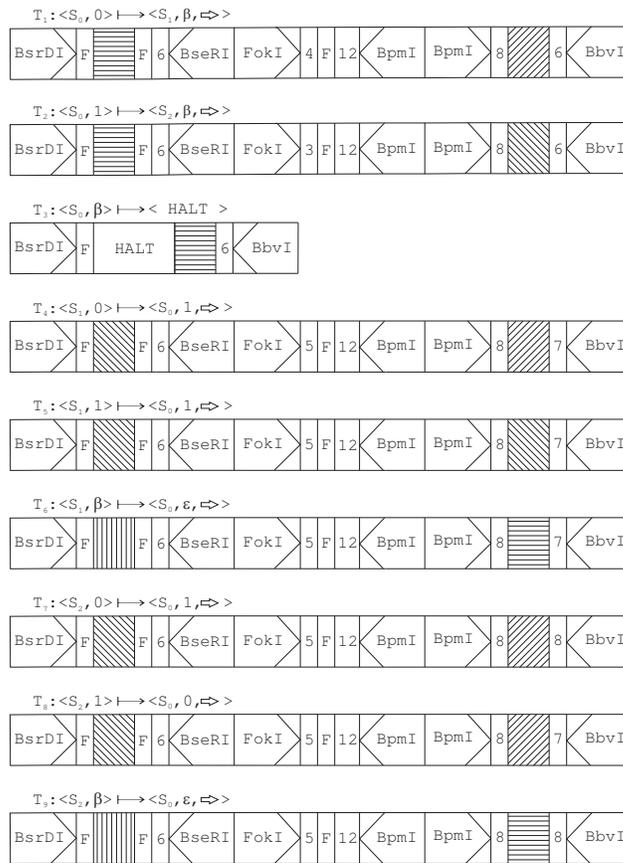
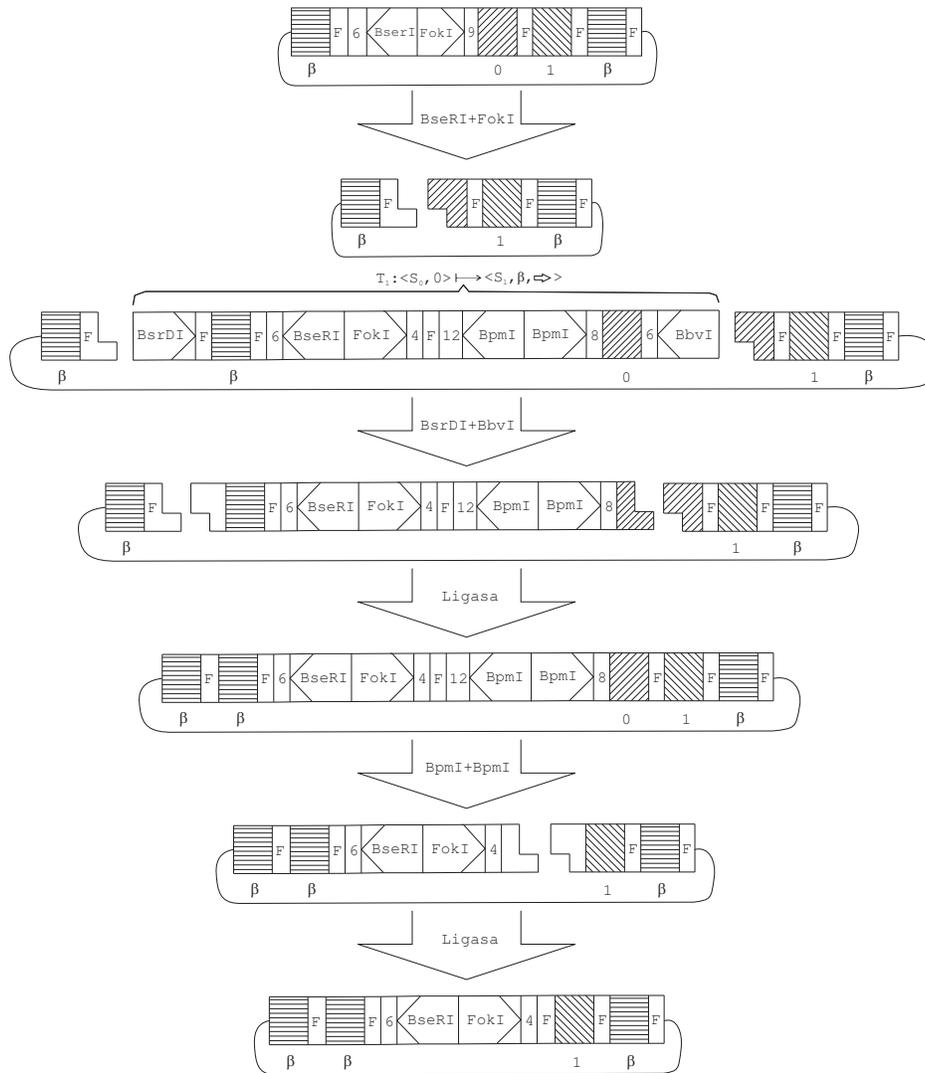
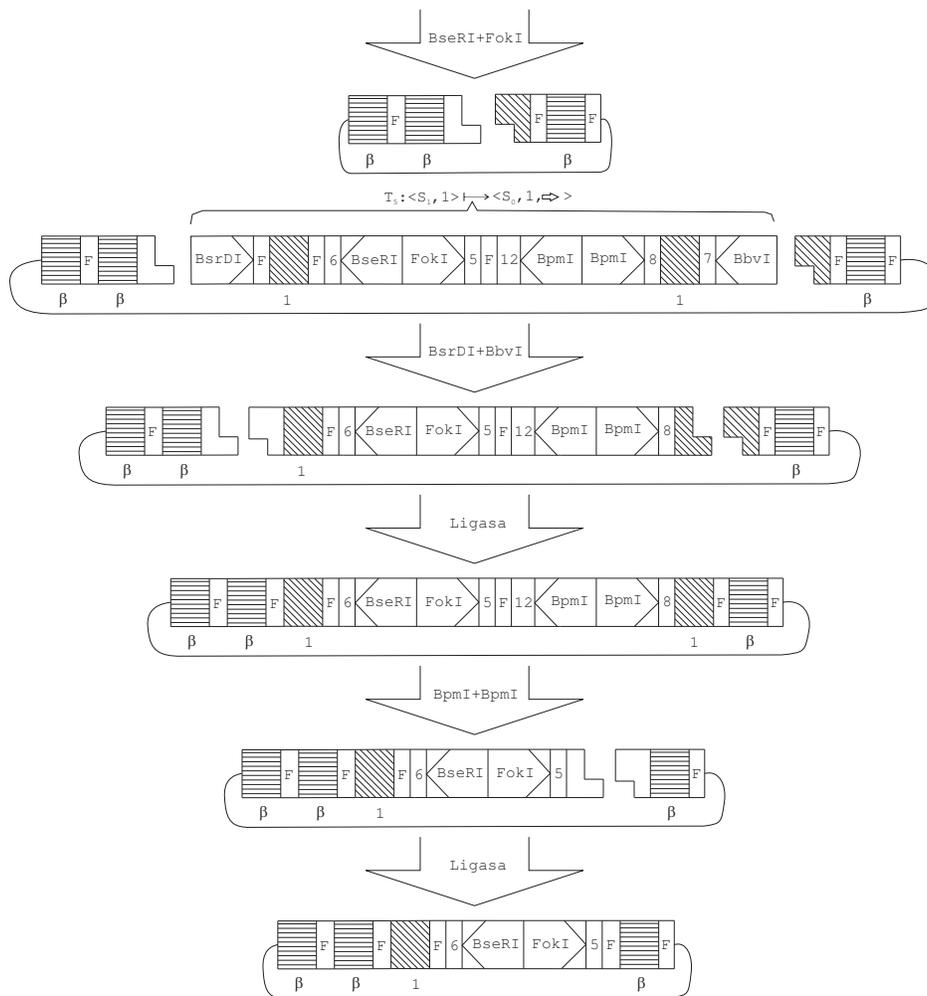


Figura 4.4: Moléculas de Transición

De esta forma, el proceso que sigue la máquina sobre la entrada recién descrita, se muestra en la siguiente figura.





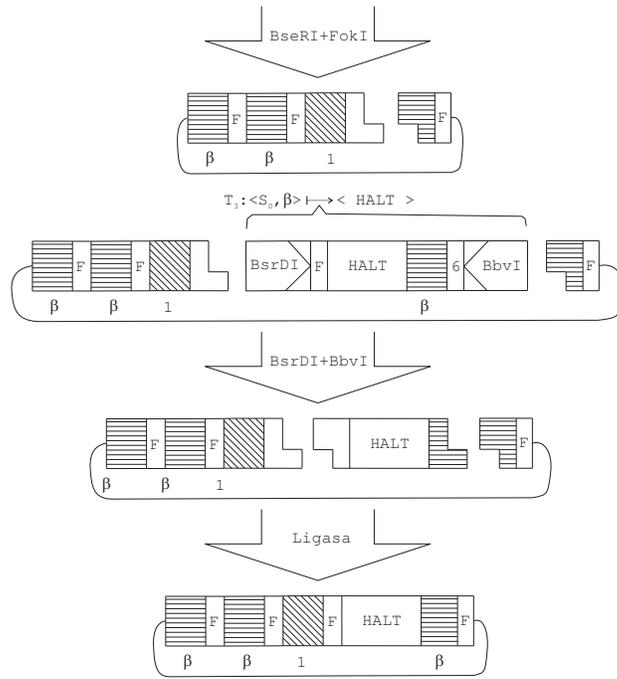


Figura 4.5: Corrida

Capítulo 5

Conclusiones

Con los resultados de los dos capítulos anteriores (3, 4) damos por terminada la exposición de la simulación de una compuerta N-AND mediante moléculas de ADN. Queda por discutir la posibilidad de su implementación real, es decir, dar el salto del *papel* al *laboratorio*. En este capítulo trataremos ese tema.

5.1. Algunas consideraciones acerca de la implementación de las máquinas de ADN

En la primera parte del Capítulo 3, la que corresponde a la simulación de $NAND_1$ con $n = 1$, el objeto $\langle \Gamma, \Delta, \Sigma, \Phi \rangle$ está constituido por conjuntos que a su vez contienen moléculas que están definidas casi totalmente. Las cadenas en el conjunto Σ están dadas explícitamente, mientras que las moléculas de los conjuntos Δ y Γ están indeterminados en los *nucleótidos espaciadores*. Debe recordarse que la elección de dichos nucleótidos tiene que ser tal que no provoquen un conflicto en el funcionamiento del autómata, como sería elegir para ellos el sitio de reconocimiento de la enzima de restricción *FokI* cuando corta de izquierda a derecha (es decir, cuando la cadena doble GGATG se encuentra en la cadena superior) o cuando corta de derecha a izquierda (es decir, cuando la cadena doble CATCC se encuentra en la cadena superior). Una simple observación basta para que nos demos cuenta de que ésto no puede ocurrir en las moléculas del conjunto Γ , no así para las del conjunto Δ , donde la libertad de colocar 100 nucleótidos espaciadores en la molécula D_1 y 50 en la molécula D_2 puede poner en peligro el funcionamiento correcto de la máquina. Por tanto, es necesario cuidar que dentro de estos nucleótidos no se incluya la subcadena que es reconocida por la enzima *FokI*.

Otro factor que se debe tomar en cuenta al computar con moléculas de ADN es la secuencia en que se realiza el cómputo, pues cada cadena de ADN contenida en el tubo de ensayo realiza su operación siguiendo una serie de pasos bien determinados. Esta serie de pasos no es otra cosa que cada uno de los cortes y uniones que se describieron en la Afirmación 3.1.1 (y posteriormente en los Teoremas 3.2.1 y 4.2.1) pero considerando una sola cadena (y no las miles que puedan tenerse en un tubo de ensayo) computando en condiciones *ideales*, es decir, descartando la influencia de otras moléculas sobre la cadena de entrada. En la aplicación, el proceso comenzaría colocando, en un tubo de ensayo, el *software* (que en este caso bien podría llamarse

wetware), el cual consta de las moléculas de transición (tal vez unas 10^{18} cadenas por molécula de transición); el *hardware*, que consiste de la enzima de restricción *FokI*, la *Ligasa*, el ATP y las moléculas de detección; y por último, la cadena de entrada en su configuración inicial (también unas 10^{18} moléculas); y una vez que se han colocado estos tres elementos en un tubo de ensayo en condiciones que simulan el interior de una célula, el cómputo se realiza *casi* inmediatamente e independientemente de la interacción con el laboratorista. En este punto podemos preguntarnos qué hay en el interior del tubo de ensayo y si podríamos esperar que todas las cadenas de entrada, que se colocaron allí en un principio, han concluido el cómputo en el mismo estado. Pero esto no ocurre así, ya que todas las moléculas en el tubo de ensayo están sometidas a la influencia de unas moléculas con otras. De tal forma que habrán cadenas de entrada que no se peguen adecuadamente con las moléculas de transición, otras que no han sido cortadas correctamente por la enzima de restricción y otras más que terminaron el proceso como era esperado; esto no debe desalentarnos, pues recordemos que el propio Adleman afirma que podemos esperar que por lo menos una de todas esas miles de moléculas computando en paralelo termine el proceso correctamente (ver Preliminares)[7]. Al final del proceso, podemos leer el resultado del cómputo mediante *gel electroforesis* (ver Preliminares) ya que las moléculas de detección tienen longitudes bien determinadas. Por ejemplo, las cadenas finales (3.27) y (3.45) de la Afirmación 3.1.1 poseen longitudes distintas: la primera de 106 bps y la segunda de 54 bps, de tal forma que la primera reporta que la máquina terminó en estado de aceptación (1), mientras que la segunda no (0).

Quando consideramos la generalización de la máquina anterior, es decir, aquella que puede recibir un número de *bytes* mayor o igual a 2, tenemos que tomar más decisiones que las que tomamos en el caso anterior ($n = 2$), pues en primer lugar debemos elegir una asignación para los símbolos en $\Sigma = \{0, 1, t\}$ codificados molecularmente; tal y como dijimos en el párrafo anterior, la elección que se haga sobre los nucleótidos que componen estos símbolos debe ser tal que no entre en conflicto con el funcionamiento de la máquina, donde esto quiere decir que se debe evitar seleccionar la subcadena reconocida por la enzima de restricción; asimismo se debe procurar que los símbolos, codificados molecularmente, difieran lo más que se pueda el uno del otro. Es decir, se deben evitar errores como el de dar una asignación a un símbolo y para los símbolos restantes modificar, digamos, una sola base del primero, ya que esto puede llevar a enlaces no deseados entre cadenas. Con respecto a las moléculas de transición y detección podemos decir lo mismo que en el párrafo anterior en relación a los nucleótidos espaciadores; especialmente en el caso de las moléculas de detección, pues como queremos leer la salida de la máquina por medio de *gel electroforesis* y cada una de estas moléculas de detección poseen un cabo pegajoso de cuatro nucleótidos, una manera de optimizar esta *lectura* es haciendo que los nucleótidos espaciadores de cada una de estas moléculas (r, x, y, z y w) difieran considerablemente. Ahora, acerca del cómputo real de esta máquina, podemos decir algo similar a lo descrito en el párrafo de arriba; i.e. que al mezclar *hardware*, *software* y la entrada, el cómputo se realiza de forma automática resultando en cadenas que lograron concretar el proceso correctamente y otras que no, donde *correctamente* significa que llevaron a cabo el proceso tal y como se describe en cada uno de los **CASOS** del Teorema 3.2.1; a saber:

1. La enzima *FokI* corta la cadena de entrada.

2. Alguna de las moléculas de transición, que *flotan* con la cadena de entrada, se *pega* a esta última dado el cabo pegajoso de ambas y gracias a la acción de la *Ligasa*.
3. Se repite el paso 1 hasta que en la cadena de entrada NO se encuentran más sitios de reconocimiento de la enzima de restricción, lo cual equivale a decir que alguna de las moléculas de detección se ha pegado a la cadena de entrada.

En el más ideal de los casos, todas las cadenas de entrada en el tubo de ensayo ejecutarán los pasos anteriores como se describe en el Teorema 3.2.1; en el más real de los casos, sólo un porcentaje de estas cadenas lo harán.

Ahora analicemos la implementación de la *Máquina de Turing* descrita en el Capítulo 4. Análogamente a la máquina para la generalización de $NAND_1$, tenemos que elegir las bases que componen cada uno de los símbolos en $\Sigma = \{0, 1, \beta, \varepsilon\}$ además de las bases de los cuatro nucleótidos del sufijo F y la cadena que representa el estado de detención $HALT$; en esta ocasión cuidando que esta elección sea lo más diferente posible entre símbolo y símbolo, y que en ninguna de estas cadenas (ni en los nucleótidos espaciadores) aparezca la subcadena reconocida (ya sea en su versión regular o en su versión especular) por las enzimas en Φ .

Una vez que se ha realizado esta elección, el proceso de cómputo para una cadena formada siguiendo la configuración inicial del Teorema 4.2.1, en situaciones *ideales* es el siguiente:

1. Las enzimas *FokI* y *BserI* cortan la cadena en sus sitios de reconocimiento respectivos. Como la cadena es un *plásmido* (i.e. está pegada por ambos extremos) no se pierde la información.
2. Se mezclan las moléculas de transición en Γ , previamente sometidas a la acción de las enzima *BbvI*.
3. En la mezcla, la enzima *BsrDI* actúa sobre las moléculas de transición haciéndolas susceptibles de adherirse a la cadena inicial.
4. La *Ligasa* se encarga de pegar las cadenas que en virtud de sus cabos, son susceptibles de aparearse.
5. Para borrar de la cadena el símbolo que acaba de ser leído se deja que la enzima *BpmI* actúe sobre la cadena.
6. Nuevamente se deja que la *Ligasa* pegue los cabos complementarios.
7. Se repite el paso 1 hasta que no haya ningún sitio de reconocimiento sobre la molécula de entrada, lo cual implica que la subcadena $HALT$ se encuentra ya en la cadena inicial.

El proceso anterior se describe ampliamente en [5], donde además su autor nos afirma que existen enzimas de restricción que difícilmente cortan cuando su sitio de reconocimiento se encuentra al final de una cadena (es el caso de las moléculas en Γ) por lo cual se agregan nucleótidos espaciadores al final (o al inicio) de la cadena.

Nuevamente debemos considerar que muchas de las moléculas contenidas en la solución no seguirán el proceso anteriormente descrito, mas el paralelismo y la densidad de estas moléculas son factores que aseguran que algunas de las cadenas iniciales realizarán el cómputo como es debido. Al final del proceso nos interesa saber el contenido de la *cinta*, esto es, los nucleótidos que se encuentran sobre la

cadena inicial; a diferencia de la máquina $NAND_1$ donde sólo interesaba el estado final del autómata. Por tanto, se debe recurrir al proceso de secuenciación (ver Preliminares) para conocer el resultado del cómputo. Como se dijo en su momento, la secuenciación es un proceso cuya complejidad aumenta conforme la longitud de la molécula aumenta; lo cual significa que cuanto más grandes sean los *bytes* a procesar, mayor será la dificultad de leer su función NAND.

5.2. Discusión

El material genético promete ser un medio de cómputo muy ventajoso por las características que se expusieron al inicio de esta tesis (*paralelismo* y *complementariedad*), ofreciendo la posibilidad de enfrentar problemas con relativamente menos esfuerzo del que le toma a la computación convencional. Prueba de esto son los resultados que ha obtenido Adleman resolviendo problemas de tipo combinatorio como el, ya descrito, Problema de la gráfica Hamiltoniana y el problema SAT; incluso se ha propuesto como estrategia para romper el algoritmo de encriptación DES[6].

Probablemente este medio de cómputo no haga mejor lo que las computadoras convencionales ya hacen. Para darle la vuelta a esta aparente desventaja, se ha propuesto un modelo de cómputo híbrido el cual combina elementos de la computación electrónica con elementos de la computación genética. Por ejemplo, se ha insinuado el uso de dispositivos mecánicos (robóticos) que lleven a cabo todos los procesos que actualmente realiza el laboratorista, como son la preparación de la mezcla de enzimas de restricción, moléculas de transición, cadenas de entrada, etc.; extracción de moléculas de la solución usando balines como en el Experimento de Adleman (ver Preliminares); enfriamiento y calentamiento de una solución para separar las cadenas complementarias de ADN; y todos los pasos para llevar a cabo las técnicas de *gel electroforesis*, PCR, secuenciación, etc. Así, se podría hablar de un cómputo que aprovecha las capacidades de ambos modelos resultando en un nuevo modelo completamente autónomo [6].

Naturalmente aún queda mucho camino que recorrer y muchos problemas que resolver. Aún no se ha investigado lo suficiente acerca de los mecanismos de detección y recuperación de errores. Un esbozo de las técnicas, que pueden ayudar a paliar los efectos indeseados durante el cómputo, se da en [5]. Podemos ubicar varios de los errores que pueden emerger durante el proceso; a continuación se describen algunos de ellos y su posible resolución.

Enlaces fallidos En ocasiones, la *Ligasa* no pegará un par de cadenas complementarias como es debido. Las cadenas que resultan de una mala acción de esta enzima, pueden ser eliminadas utilizando una exonucleasa.

Enlaces erróneos Otro problema es que la *Ligasa* pegue cadenas que no deberían pegarse, esto es, que no sean complementarias. Esta situación ocurre en determinadas condiciones, y las cadenas que resultan pueden ser detectadas usando la enzima *S1* (ver Introducción) la cual es capaz de cortar subcadenas simples en el interior de una cadena doble. Una vez hecha esta detección se puede emplear una exonucleasa para eliminar la cadena incorrecta.

Cortes fallidos Los cortes efectuados por las enzimas de restricción no son siempre exactos. En algunos casos, durante la ejecución de la *Máquina de Turing*

algunas cadenas de ADN no se deshacen completamente de sus sitios de reconocimiento como se espera, y así pueden entrar en conflicto con el resto de la operación. Así como es posible agregar moléculas en la secuencia `HALT` de la *Máquina de Turing* para detectar aquellas cadenas que hayan concluido el cómputo, se puede emplear cierto tipo de moléculas para etiquetar aquellas cadenas que, tras la acción de una enzima de restricción, aún contengan el sitio de reconocimiento de la enzima.

Cortes erróneos Puede ocurrir, aunque es muy raro, que una enzima de restricción corte donde no debe. Esta situación tiene lugar cuando la enzima se somete a condiciones de operación anormales, y generalmente el corte se lleva a cabo sobre subcadenas que se asemejan a su sitio de reconocimiento. Se recomienda entonces, llevar a cabo las reacciones de restricción (i.e. acción de una enzima de restricción) bajo las condiciones normales de cada enzima, así como también elegir nucleótidos que se distingan de los sitios de reconocimiento.

El estudio e implementación de la detección y corrección de errores es un tema de enorme importancia para el desarrollo de este modelo de cómputo. Una descripción exhaustiva de esta teoría está fuera de los alcances de esta tesis, pero bien podría ser el tema de otra, en la cual sumen esfuerzos la biología, la química, la matemática y las ciencias de la computación.

5.3. Reflexión final

Es momento de una reflexión final. La pregunta es qué sería de la *Máquina de Turing* si omitimos una enzima del conjunto Φ . ¿Podríamos aún construir una máquina con estas características? Supongamos que no tenemos la enzima de restricción *FokI*. ¿Podríamos todavía crear esta *Máquina de Turing*? ¿Siquiera el autómata por el cual resolvimos la función $NAND_1$? Supongamos que de hecho NO tenemos ninguna de estas enzimas. ¿Qué será de la computación genética? ¿Adleman podría todavía realizar su Experimento? Suena difícil hacerlo. Parecer ser que el uso de las enzimas de restricción es imprescindible en la computación genética, y que es en virtud de éstas que ponemos los ojos en ella. ¿Será? ¿Podríamos afirmar lo mismo de las *otras* características del material genético? Podríamos, claro, pero estas características se conocen en el momento en que se estudia al Hombre mismo, mientras que el descubrimiento de las enzimas de restricción se da cuando se pone atención a *otros* organismos (v.gr. las bacterias); y bien podría ocurrir que en uno de estos organismos se esconda una enzima de restricción que ayuda a resolver eficientemente un problema en particular.

Ocurre que el estudio de las enzimas de restricción marca el paso de la computación genética. Por ejemplo, se sabe que antes de que Rothemund propusiera su versión de la *Máquina de Turing* molecular, el investigador Charles Bennett había diseñado una de estas máquinas una década antes del Experimento de Adleman[5], pero su trabajo hacía referencia a enzimas inexistentes. Incluso, Benenson y Shapiro parecen concordar con Adleman cuando le citan en [4]:

En el futuro, la investigación en biología molecular proveerá técnicas mejoradas para la manipulación de macromoléculas.

Y en relación al tema de esta reflexión:

La investigación en química permitirá el desarrollo de enzimas sintéticas prediseñadas. Se puede imaginar el surgimiento eventual de una computadora de propósito general que consiste únicamente de una sola macromolécula unida a una colección ribosómica de enzimas que actúan sobre ella.

Al parecer la existencia de estas enzimas de restricción en verdad determina el poder de cómputo de este tipo de máquinas. ¿Cómo así? Podemos hablar de dos grandes instrucciones que están detrás de la discusión de los dos capítulos anteriores, a saber: *cortar y pegar* cadenas de ADN; obviamente, estas dos operaciones no actúan conforme nuestra voluntad, sino que llevan a cabo su función a través de las reglas que hemos discutido a lo largo de este trabajo. (La unión de dos cadenas se consigue siempre y cuando éstas sean complementarias. El corte de cadenas se logra si en la cadena doble está contenida una subcadena de ciertas características.) Ya mencionamos que la capacidad que tienen las cadenas para aparearse, es una propiedad que tenemos desde que pusimos nuestros ojos en el material genético; en cambio, la operación de cortar cadenas (y todas las acciones que pueden tener las exonucleasas y endonucleasas sobre el material genético) extienden el panorama de este modelo de cómputo; se trata de una suerte de *caja de herramientas*, y como tal, la ausencia de una herramienta puede repercutir considerablemente en el producto final.

Así, el autómatas más poderoso que puede ser imaginado (una *Máquina de Turing*) está condicionado, en lo que toca a la computación genética, a la disponibilidad y/o existencia de cierto tipo de moléculas. Pero esto no debe de sorprendernos. Ya en la historia de la computación electrónica se da cuenta del esfuerzo que implicaba construir una computadora cuyo principal componente eran las válvulas termiónicas (tubo de vacío) y cómo el advenimiento de los transistores ha logrado llevar la computación hasta el punto donde se encuentra actualmente, sustituyendo los complicados y fácilmente incandescentes tubos de vacío en la construcción de las computadoras. Entonces podría decirse que implementar una *Máquina de Turing* electrónicamente también dependía en gran medida de los materiales disponibles y, por ende, de las operaciones (o instrucciones) que podían llevarse a cabo con estos. Por ejemplo, la *Máquina Universal de Turing* más popular, la computadora personal (PC), está constituida por pequeñas pero abundantes compuertas lógicas cuya sinergia resulta en las intrincadas operaciones aritmético-lógicas de esta máquina¹. (Donde quizá la compuerta lógica más *conveniente* sea aquella que computa la función *NAND* de dos entradas binarias.) ¿Qué pasaría si prescindimos de alguno de estos dispositivos? ¿Sería muy atrevido afirmar, en este punto de la discusión, que al parecer eso de la computación depende mucho de las operaciones (instrucciones) que se permiten realizar sobre un conjunto de símbolos o corrientes eléctricas o nucleótidos?

¿Qué habría imaginado Turing si no hubiera concebido la idea de una *cabeza* que *lee y escribe* sobre una cinta infinita? Como puede verse, la noción de *Máquina de Turing* también depende de ciertas operaciones; a saber: *leer* sobre una cinta, *escribir* sobre una cinta, y finalmente, *mover* esta cabeza de *lectura/escritura* hacia

¹En sentido estricto ésta no es una *Máquina de Turing* ya que carece de una cinta infinita, aunque siempre podemos hacer crecer la memoria *RAM* de estas máquinas, y cuando esto no es posible se debe confiar en la gestión del sistema operativo.

la izquierda o hacia la derecha. Si bien matemáticamente una *Máquina de Turing* está definida en términos de una función, no debe olvidarse que incluso esta concepción está sujeta a la existencia de ciertas operaciones por las cuales dicha definición no es completamente mecánica. Por ejemplo, consideremos la máquina del Capítulo 4, definida como el alfabeto:

$$\Sigma = \{0, 1, \varepsilon, \beta\}$$

El conjunto de estados:

$$Q = \{S_0, S_1, S_2, HALT\}$$

Y las transiciones:

$$\begin{aligned} < S_0, 0 > \mapsto < S_1, \beta, \Rightarrow > \\ < S_0, 1 > \mapsto < S_2, \beta, \Rightarrow > \\ < S_0, \beta > \mapsto < HALT > \\ < S_1, 0 > \mapsto < S_0, 1, \Rightarrow > \\ < S_1, 1 > \mapsto < S_0, 1, \Rightarrow > \\ < S_1, \beta > \mapsto < S_0, \varepsilon, \Rightarrow > \\ < S_2, 0 > \mapsto < S_0, 1, \Rightarrow > \\ < S_2, 1 > \mapsto < S_0, 0, \Rightarrow > \\ < S_2, \beta > \mapsto < S_0, \varepsilon, \Rightarrow > \end{aligned}$$

Dicha definición no es para nada mecánica, sino que depende totalmente de que el lector comprenda los conceptos de *cambio de estado*, *escritura y lectura de un símbolo* y por supuesto de la capacidad del lector de imaginar una *cinta* sobre la cual se encuentra con símbolos del alfabeto conforme recorre su extensión. ¿Pues qué haría un individuo a quien se presentase esta máquina? Lo primero que haría sería preguntar por la *cinta*, o mejor dicho, por el contenido de la *cinta*, la cual no necesariamente debe verse como tal, sino que basta con que se le otorgue una cadena de símbolos del alfabeto y él comenzará a procesar dicha cadena siguiendo las reglas anteriores; guardando, en todo momento, registro del estado actual del proceso, siendo S_0 el estado inicial. ¿Qué operaciones básicas realizó este individuo al final del cómputo, es decir, cuando se encuentra en el estado **HALT**? ¿Qué instrucciones son requisito comprender? Básicamente la acción de *leer* un carácter de la cadena inicial, *buscar* en las reglas de transición la combinación de carácter leído y estado actual, donde saber el estado actual supone su *lectura* en algún sitio, *escribir* el nuevo carácter sobre la cadena inicial, *guardar* registro del nuevo estado y, en caso de que el nuevo estado no sea el estado **HALT**, *leer* el siguiente carácter a la izquierda o a la derecha del carácter recién escrito, según lo diga la regla de transición. Y he aquí la afirmación más contundente: sin alguna de estas instrucciones escritas *en cursivas* la noción de *Máquina de Turing* sería distinta a como la conocemos actualmente.

Lo mismo puede decirse de las versiones electrónicas y moleculares del *Autómata Determinista Finito* y la *Máquina de Turing*. Si omitimos la acción de algún dispositivo electrónico (v.gr. el *apuntador de programa*) o de alguna enzima de restricción (v.gr. *FokI*) el cómputo de estas máquinas se vuelve muy diferente,

al grado de que quizá no exista tal.

Como puede verse, las preguntas surgidas en esta sección son de mucho interés para nuestra noción de computabilidad. Un trabajo interesante sería el que se dedicara a responder qué tanto se ven afectadas éstas máquinas moleculares al desestimar la acción de alguna enzima (ya sea para extender o para acortar una cadena de ADN).

Una cuestión aún más importante es la de promover un estudio interdisciplinario para el desarrollo de este modelo de cómputo, que promete ser un nuevo paradigma en la lucha contra diversas enfermedades crónico-degenerativas; así como también una nueva manera de enfrentar los problemas difíciles de la computación actual. Este trabajo se dedicó únicamente a resolver una *pequeña* parte de los problemas que aún debe resolver la computación genética para poder erigirse como un modelo total de cómputo, pero se puede decir que se hizo *algo* si tras su lectura nos sentimos motivados por el estudio de esta teoría desde todos sus frentes: *biológico*, *matemático*, *químico*, *computacional*, y por qué no decirlo, *filosófico*.

Bibliografía

- [1] Vineet Gupta, Srinivasan Parthasarathy, Mohammed J. Zaki. *Arithmetic and Logic Operations with DNA*. Technical Report. UMI Order Number: TR650. University of Rochester. 1997
- [2] Yaakov Benenson, Tamar Paz-Elizur, Rivka Adar, Ehud Keinan, Zvi Livneh, Ehud Shapiro. *Programmable and autonomous computing machines made of biomolecules*. NATURE, Vol. 414, pp. 430-434. Noviembre 2001.
- [3] Yaakov Benenson, Binyamin Gil, Uri Ben-Dor, Rivka Adar, Ehud Shapiro. *An autonomous molecular computer for logical control of gene expression*. NATURE. Vol. 429, pp. 1-6. Mayo 2004.
- [4] Yaakov Benenson, Ehud Shapiro. *Bringing DNA Computers to life*. Scientific American Special Edition, Vol. 17. No. 3. Septiembre 2007
- [5] Paul Wilhelm Karl Rothemund. *A DNA and restriction enzyme implementation of Turing Machines*. <http://www.ugcs.caltech.edu/~pwkr/oett.html>
- [6] Gheorghe Păun, Grzegorz Rozenberg, Arto Salomaa. *DNA Computing. New computing paradigms*. Springer. 1998. ISBN 3-540-64196-3
- [7] Leonard M. Adleman *Computing with DNA*. Scientific American, pp. 54-61. Agosto 1998.