

Universidad Autónoma de la Ciudad de México.

Proyecto Final:

Propagación de un prion demostrado por un Autómata.

Teoría de la Computación.

Semestre 2024 – I.

Profesora: Elizabeth López Lozada.

Integrantes:

Francisco Javier Pérez Salamanca.

20-011-0026

Leonardo Valdes Palafox.

18-011-0249

Grupo 1101.

Fecha de entrega: 12/05/2024.

Índice.

Contenido del proyecto.

| | |
|---|----|
| Resumen. | 3 |
| Introducción. | 4 |
| Trabajos Relacionados. | 5 |
| Problema y Solución. | 7 |
| Complejidad del algoritmo. | 15 |
| Discusión. | 16 |
| Conclusión. | 17 |
| Bibliografía. | 18 |

Resumen.

El proyecto busca la demostración sencilla de la creación de un prion y su propagación a otras proteínas sanas para la posterior infección de enfermedades.

Mediante los autómatas y la investigación de los priones, se da una aproximación de los pasos y fenómenos que llevan a la creación e infección de priones que causan enfermedades a distintos tipos de animales incluyendo a los seres humanos, uno de estos mostrará de forma simplificada como es que un prion se crea y otro ejemplificará una enfermedad causada por priones y sus consecuencias en el cuerpo humano.

Introducción.

Los autómatas son modelos matemáticos que se utilizan para describir sistemas con un número finito de estados y transiciones entre esos estados. Son especialmente útiles para modelar sistemas que tienen un comportamiento predecible basado en su estado actual y las entradas que reciben.

Los priones son los agentes causantes de un grupo de patologías neurodegenerativas mortales característicos de los mamíferos, también conocidas como encefalopatías espongiformes transmisibles.

En el contexto de los priones, los autómatas pueden ser útiles para modelar el comportamiento de estas proteínas infecciosas. Un prion puede existir en un estado normal o en un estado infeccioso, y puede cambiar entre estos estados bajo ciertas condiciones. Además, un prion infeccioso puede interactuar con otras proteínas y convertirlas en priones infecciosos, lo que puede llevar a la propagación de enfermedades.

Trabajos Relacionados.

Tras nuestra investigación a los priones y su comportamiento definido por autómatas, no encontramos un trabajo exactamente relacionado a los priones, sin embargo, sí que hay investigaciones y demostraciones con autómatas sobre la propagación de enfermedades.

En la actualidad se utilizan diversos tipos de modelados para simular la dispersión de diversos cuadros infecciosos, esta labor es de vital importancia ya que ayudan a prevenir grandes cantidades de contagios y decesos tal como ya ha ocurrido con algunas pandemias como pueden serlo la gripe española (1918-1920), la gripe asiática (1957) entre otras.

Uno de los modelos más ampliamente usados es el conocido como modelo “SIR” por sus siglas en inglés que significan Susceptibles (susceptibles), Infectives (infectivo ó infectado) y Recovered (recuperados con inmunidad) y es constantemente formulado como se muestra en la siguiente imagen:

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta IS}{N}, S(0) = S_0 \geq 0, \\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I, I(0) = I_0 \geq 0, \\ \frac{dR}{dt} &= \gamma I, R(0) = R_0 \geq 0,\end{aligned}$$

donde $S(t)$, $I(t)$, $R(t)$ son los números dados por la clase modelo de maneta que $S(t)+I(t)+R(t)=N$ y N es el número total de individuos, β la tasa de contacto y $1/\gamma$ el tiempo promedio que dura la infección a pesar de todos estos datos el modelo “SIR” clásico no ha sido suficiente para adaptar todas las variables como lo son muerte natural, emigración, inmunidad pasiva etc. De manera que ha tenido que ser adaptado a más clases y parámetros.

Por lo cual uno de los problemas al usar el modelo SIR clásico ó modelo SIR extendido es la incapacidad de los mismos para proporcionar datos demográficos y geológicos, en su lugar es preferible usar Autómatas Celulares (CA por sus siglas en inglés) en estos sistemas el tiempo y el espacio son discretos, en este modelo con un enrejado de condiciones y estados finitos, un conjunto finito de células que define la interacción vecina y está definido como una 8tupla.

$CA = (Z, N, \Sigma, Q, \delta, \sigma, Q_0, F),$

donde

$Z = \text{configuración de la célula}$

$N = \{n_1, n_2, \dots, n_d, \dots\},$

$\Sigma = \{a_1, a_2, \dots, a_m\},$

$Q = \{q_0, q_1, \dots, q_q\},$

$\delta: Q \times Q_{n_i} \rightarrow Q, n_i \in N,$

$\sigma: Q \rightarrow \Sigma,$

$Q_0 \subseteq Q,$

$F \subseteq Q.$

N representa la célula vecina Σ el alfabeto de entrada, Q es el número finito de estados, δ es el estado de transición de función, σ es la función de salida, Q_0 el estado final y F el estado final.

En términos muy generales y para simplificar existen varios autómatas, algunos de ellos programados en java los cuales sirven para simular los procesos infecciosos de virus, enfermedades y pandemias, dichos autómatas han sido modelados matemáticamente y programados en lenguajes como java, su funcionamiento es cuando menos peculiar. Por ejemplo, uno de ellos está constituido por cinco clases.

La primera: Autómata celular: Esta clase es una clase abstracta que define al autómata celular necesita llamar a un método "llamado compute ()" para generar una simulación, la función principal de este método es la de iterar a través de todas las células dentro del autómata

La segunda: Célula: Esta clase abstracta genera los comportamientos de las células dentro del autómata, cada célula representa áreas geográficas que pueden tener sus propios comportamientos

La tercera: Propiedades de la célula: Esta clase define las propiedades de la célula, las cuales pueden ser adaptadas para modelos específicos.

La cuarta: Dimensión del autómata: Permite definir un autómata en varias dimensiones(1d,2d,3d)

Y finalmente: Coordenadas: Esta clase permite calcular las coordenadas que ocupa cada célula, las coordenadas se almacenan en un arreglo de 1 dimensión, pero se les asigna posiciones de variables x, y z.

Problema y Solución.

Se propone demostrar el comportamiento de un prion infeccioso y su propagación a proteínas saludables a través de un autómata, esto mediante un dibujo explicativo sobre el diagrama y un programa en lenguaje C que muestre esa secuencia de infección.

Primero expliquemos brevemente qué es un prion y cómo es que infecta a proteínas saludables:

Un prion es una proteína mal plegada capaz de transmitir su forma mal plegada a otras variedades de la misma proteína. El proceso de creación de un prion es cuando una serie de aminoácidos, que crean a la proteína, se juntan de manera que estos se pliegan de formas únicas ya que dependen meramente de su secuencia de aminoácidos. Estos priones son denominados PRPc (proteína priónica celular) y realizan su trabajo con normalidad como: formar estructuras que son indispensables para las células, reacciones químicas que ayudan contra infecciones, etc. Sin embargo y paradójicamente, cuando algo sucede en este pliegue de cadenas de aminoácidos y este sale mal, o de forma incorrecta, entonces se genera un PRPsc (mismas siglas con el añadido de s que es 'scrapie'). Esta proteína malformada no cumple su función y puede contagiar a otras proteínas saludables provocando la misma mala función. Es entonces cuando se crean grupos de estos priones malignos y se juntan en grandes números provocando daños a los tejidos y su principal peligro, daños muy graves al sistema nervioso central. Gracias a estas conglomeraciones de grupos de PRPsc el cuerpo humano es incapaz de erradicarlos, por lo que provocan enfermedades letales y degenerativas con el paso del tiempo. Una de las enfermedades más representativas es la del Kuru (que trataremos más adelante) y la enfermedad de las vacas locas, por sus siglas EEF (encefalitis espongiforme transmisible), este nombre debido a:

- Encefalitis: Una afectación al sistema nervioso.
- Espongiformes: Daños tan graves al cerebro que provoca "hoyos" que le dan una apariencia de esponja.
- Transmisible: Que se puede transferir a diferentes sistemas.

Dicho lo anterior, entonces su busca dar una demostración sencilla y clara sobre la transformación e infección de estos priones dañinos.

1.- Demostración del contagio de un Prion a través de un autómata.

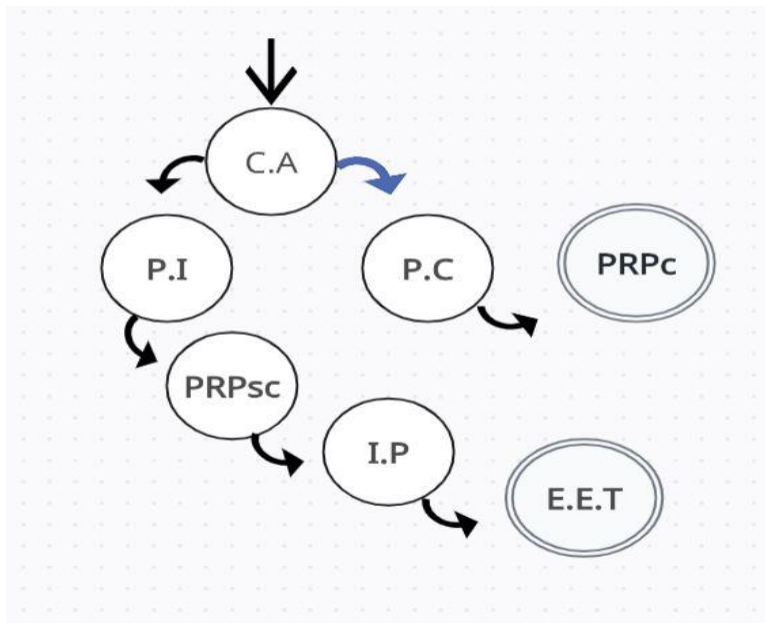


Figura 1.

Aquí tenemos una imagen del autómata creado para ejemplificar la infección de un prion (figura 1).

Definamos nuestros estados, siendo:

- CA: Creación de **C**adenas de **A**minoácidos.
- PC: Es el **P**liegue **C**orrecto de las cadenas.
- PRPc: Es la transformación exitosa del prion funcional, estado final.
- PI: **P**liegue **I**ncorrecto, entonces empiezan la transformación defectuosa del prion.
- IP: **I**nteracción con otras **P**roteínas provocando su infección
- PRPsc: La transformación de los priones a priones malignos.
- EET: Es ya la propagación de estos grupos de infección provocando enfermedades, en este caso es la **E**ncefalitis **E**spongiforme **T**ransmisible, estado final.

Esto nos quiere decir la siguiente secuencia:

1. El estado inicial CA es el pliegue correcto de estas cadenas de aminoácidos, dependiendo de su correcto o incorrecto procedimiento pasaremos a PI O PC.
2. En caso de que el procedimiento salga bien avanzamos a el estado de pliegue correcto y continuamos hasta la modificación del prion a PRPc.
3. Por otro lado, si avanzamos al estado de pliegue incorrecto entonces empieza la mutación del prion saludable a uno defectuoso.

4. Tras el inicio de la mutación, el prion empieza su cambio al estado de un PRPsc, donde su forma defectuosa está completada
5. Una vez el prion se mal formó, avanza al estado de interacción y comienza a relacionarse con otras proteínas sanas mutándolas a una forma errónea como la suya, entonces crean grupos de estos priones que se propagarán en el sistema del ser vivo.
6. Cuando los grupos de priones PRPsc se empiezan a propagar crean estragos graves en el sistema nervioso y provocan enfermedades degenerativas, en el caso del autómatas se avanza al estado de EET (encefalitis espongiforme transmisible), o más conocida como la enfermedad de las vacas locas. Entonces la enfermedad comienza.

Una vez dada la representación gráfica del autómatas y su explicación, vayamos al código en lenguaje C que lo modela.

1.1.- Código en C para el autómatas del prion.

Este código ejemplifica la parte gráfica del autómatas y sigue su secuencia dependiendo a qué estado es al que se mueve. Para este particular caso y debido a que el por qué del mal plegamiento de las cadenas sucede, se dejó al azar con la función random entre dos números, siendo uno el correcto pliegue y el otro el mal pliegue.

```
main.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  // Definición de los estados
6  enum Estado {S1, S2, S3, S4, S5, S6, S7};
7
8  // Función para manejar el plegamiento
9  enum Estado plegamiento(enum Estado estado_actual, int plegamiento_correcto) {
10     if (estado_actual == S1) {
11         if (plegamiento_correcto) {
12             return S2;
13         } else {
14             return S4;
15         }
16     }
17     // Si no estamos en el estado S1, no hacemos nada
18     return estado_actual;
19 }
20
21 // Función para manejar la interacción con otras proteínas
22 enum Estado interaccion(enum Estado estado_actual) {
23     if (estado_actual == S2) {
24         return S3;
25     } else if (estado_actual == S4) {
```

```

main.c
25 } else if (estado_actual == S4) {
26     return S5;
27 }
28 // Si no estamos en el estado S2 o S4, no hacemos nada
29 return estado_actual;
30 }
31
32 // Función para manejar la propagación
33 enum Estado propagacion(enum Estado estado_actual) {
34     if (estado_actual == S5) {
35         return S6;
36     } else if (estado_actual == S6) {
37         return S7;
38     }
39     // Si no estamos en el estado S5 o S6, no hacemos nada
40     return estado_actual;
41 }
42
43 int main() {
44
45     printf("\n Simulador de Prion. \n");
46     printf("\n Tras la formacion de una cadena de aminoacidos, el pliegue fue...\n");
47     enum Estado estado_actual = S1;
48     int plegamiento_correcto;
49

```

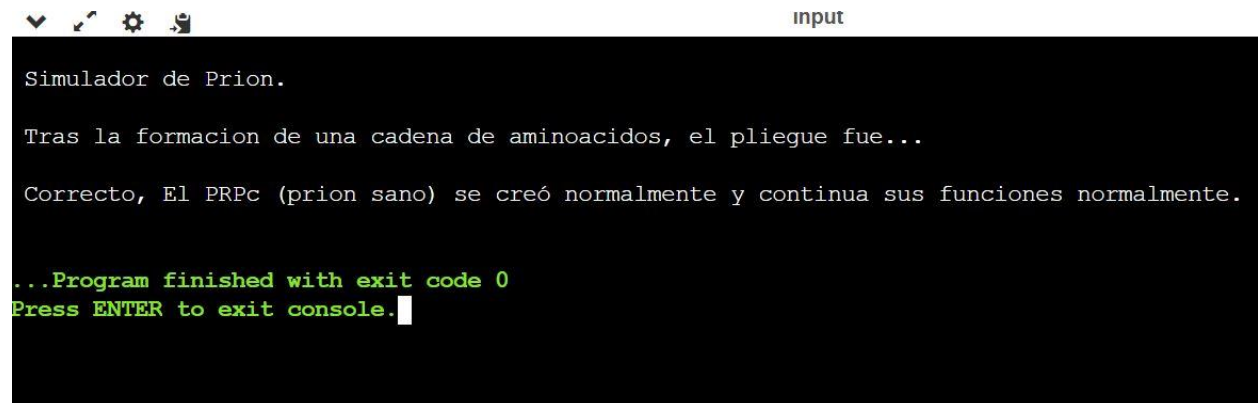
```

main.c
44
45     printf("\n Simulador de Prion. \n");
46     printf("\n Tras la formacion de una cadena de aminoacidos, el pliegue fue...\n");
47     enum Estado estado_actual = S1;
48     int plegamiento_correcto;
49
50     // Inicializamos la semilla del generador de números aleatorios
51     srand(time(NULL));
52
53     // Generamos un número aleatorio entre 0 y 1
54     plegamiento_correcto = rand() % 2;
55
56     estado_actual = plegamiento(estado_actual, plegamiento_correcto);
57
58     if (estado_actual == S2) {
59         printf("\n Correcto, El PRPc (prion sano) se creó normalmente y continua sus funciones normalmente
60     } else if (estado_actual == S4) {
61         estado_actual = interaccion(estado_actual);
62         estado_actual = propagacion(estado_actual);
63         printf("\n Incorrecto, se forma un PRPsc (prion infeccioso),\n ahora interactua con otras proteina
64     }
65
66     return 0;
67 }
68

```

Este código sigue la secuencia de los autómatas y dependiendo el valor en la entrada, arrojará lo que le sucede a las cadenas simulando la transformación real de un prion a uno maligno.

Ahora ejecutamos el código para saber qué le pasó a nuestra cadena de aminoácidos.



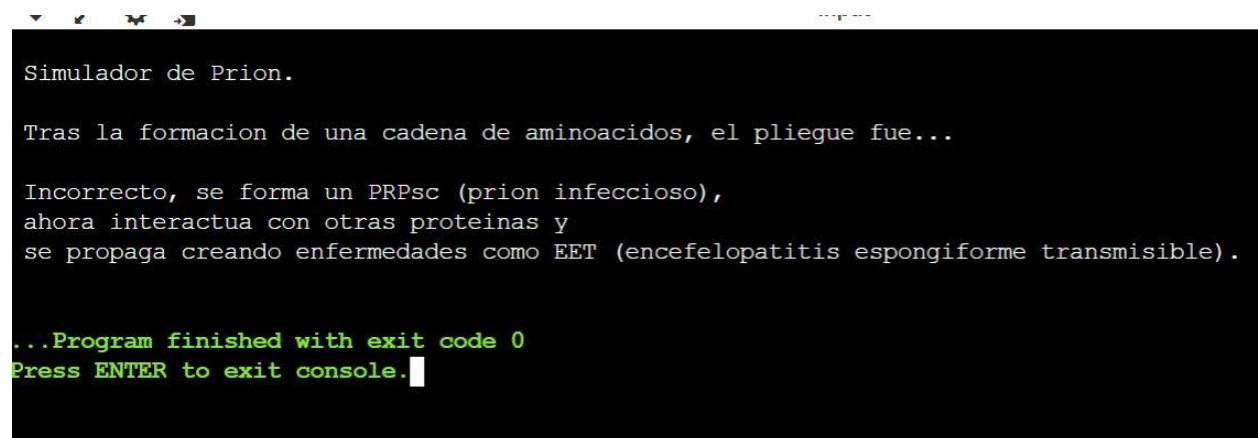
```
input
Simulador de Prion.

Tras la formacion de una cadena de aminoacidos, el pliegue fue...

Correcto, El PRPc (prion sano) se creó normalmente y continua sus funciones normalmente.

...Program finished with exit code 0
Press ENTER to exit console.
```

En éste caso el pliegue fue correcto, por lo que la proteína continuará con su función normal en el sistema.



```
input
Simulador de Prion.

Tras la formacion de una cadena de aminoacidos, el pliegue fue...

Incorrecto, se forma un PRPsc (prion infeccioso),
ahora interactua con otras proteinas y
se propaga creando enfermedades como EET (encefalopatitis espongiforme transmisible).

...Program finished with exit code 0
Press ENTER to exit console.
```

Por otro lado, si el pliegue fue malo, entonces el programa seguirá la secuencia del autómata para la transformación a un PRPRsc, agruparse y causar enfermedades.

2.- Demostración de la enfermedad del Kuru a través de un Autómata.

Primero hablemos sobre qué es la enfermedad el Kuru:

El Kuru es una enfermedad extremadamente rara. La enfermedad es causada por una proteína infecciosa (prión) que se encuentra en el tejido cerebral humano contaminado y fue descubierta en personas de Nueva Guinea que practicaban una forma de canibalismo, comiéndose los cerebros de los muertos como parte de los rituales funerarios. Esta práctica fue abolida en 1960, pero debido al largo período de incubación del kuru, se siguieron reportando casos durante muchos años.

Cool provoca cambios en el sistema nervioso y el cerebro similares a la enfermedad de Creutzfeldt-Jakob. Enfermedades similares ocurren en el ganado, incluida la encefalopatía espongiforme bovina (EEB), también conocida como enfermedad de las vacas locas.

Entonces una vez sepamos que es el kuru, veamos el autómata que representa la infección del kuru.

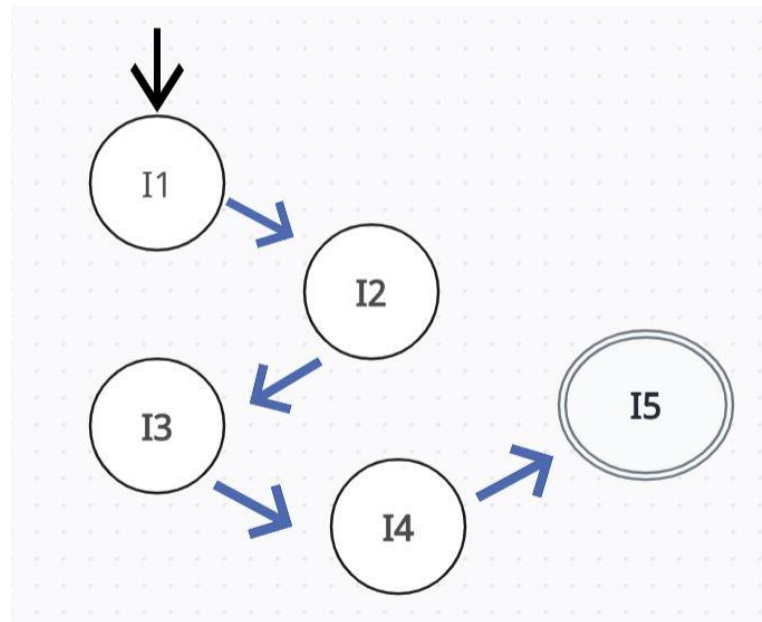


Figura 2.

Aquí tenemos una imagen del autómata creado para ejemplificar la infección del Kuru a una persona (figura 2).

Ahora definamos nuestros estados y su secuencia:

- I1: Siendo el estado de la persona sana sin enfermedades por priones.
- I2: Avanzamos al estado de la exposición (canibalismo) a la enfermedad y esta comienza su secuencia.
- I3: Es el estado del avance de la enfermedad transformando sus proteínas a priones malignos en el cerebro, provoca daños graves al sistema nervioso central.
- I4: El estado posterior al periodo de incubación, se muestran los síntomas en la persona infectada.
- I5: El estado final, cuando la enfermedad causa estragos letales y el paciente muere.

2.2.- Una vez explicados los estados del autómata y sus secuencias, vayamos al código.

```
1
2 #include <stdio.h>
3
4 // Definición de los estados
5 enum Estado {S1, S2, S3, S4, S5};
6
7 // Función para manejar la exposición
8 enum Estado exposicion(enum Estado estado_actual) {
9     if (estado_actual == S1) {
10         return S2;
11     }
12     // Si no estamos en el estado S1, no hacemos nada
13     return estado_actual;
14 }
15
16 // Función para manejar la migración de priones al cerebro
17 enum Estado migracion(enum Estado estado_actual) {
18     if (estado_actual == S2) {
19         return S3;
20     }
21     // Si no estamos en el estado S2, no hacemos nada
22     return estado_actual;
23 }
```

```

24
25 // Función para manejar el desarrollo de síntomas
26 enum Estado desarrollo_sintomas(enum Estado estado_actual) {
27     if (estado_actual == S3) {
28         return S4;
29     }
30     // Si no estamos en el estado S3, no hacemos nada
31     return estado_actual;
32 }
33
34 // Función para manejar la progresión de la enfermedad
35 enum Estado progresion_enfermedad(enum Estado estado_actual) {
36     if (estado_actual == S4) {
37         return S5;
38     }
39     // Si no estamos en el estado S4, no hacemos nada
40     return estado_actual;
41 }
42
43 int main() {
44     enum Estado estado_actual = S1;
45
46     printf("\n Simulador de Infeccion por Kuru. \n");
47

```

```

44     enum Estado estado_actual = S1;
45
46     printf("\n Simulador de Infeccion por Kuru. \n");
47
48     estado_actual = exposicion(estado_actual);
49
50     if (estado_actual == S1) {
51         printf("\n La persona no fue expuesta a priones, por lo tanto no mostrará síntomas.\n");
52     } else if (estado_actual == S2) {
53         estado_actual = migracion(estado_actual);
54         estado_actual = desarrollo_sintomas(estado_actual);
55         estado_actual = progresion_enfermedad(estado_actual);
56         printf("\n La persona fue expuesta a priones por ritual de consumo de carne humana.\n");
57         printf("\n Muestra síntomas como dolor en brazos y piernas, temblores musculares, problemas de coordinación.\n");
58         printf("\n El kuru puede provocar episodios de locura y risa maniaca que se vuelve incontrolable, a veces homicida.\n");
59         printf("\n No existe tratamiento para el Kuru.\n");
60         printf("\n El infectado muere después de meses tras la aparición de los síntomas.\n");
61     }
62
63     return 0;
64 }
65
66

```


Ahora veamos el resultado de nuestro programa.

Es importante señalar que se da por hecho que la persona estará infectada y seguirá la secuencia del autómata y por ende de la enfermedad, de otra manera no tendría sentido si no hay exposición

```
Simulador de Infeccion por Kuru.  
  
La persona fue expuesta a priones por ritual de consumo de carne humana.  
  
Muestra sintomas como dolor en brazos y piernas, temblores musculares, problemas de coordinación,  
dificultad al tragar, úlceras, dolor de cabeza.  
  
El kuru puede provocar episodios de locura y risa maniaca que se vuelve incontrolable, así como depresión.  
  
No existe tratamiento para el Kuru.  
  
El infectado muere despues de meses tras la aparición de los sintomas.  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

En la salida se explica el proceso de la enfermedad, la exposición al consumo de cerebros humanos, el muestreo de los síntomas y la explicación de los mismos, y por último la muerte del paciente.

Complejidad del algoritmo.

Debido a que la representación de la mutación de los priones y la enfermedad del Kuru son sencillas con el propósito de ser claras, **su tiempo de ejecución y complejidad computacional es constante, $O(1)$.**

Al ser ejemplos sencillos y muy generalizados, su complejidad no crece debido a que no se toman en cuenta otros factores como el tiempo de mutación y propagación de los priones, así como las características específicas de cada sistema del anfitrión afectado. Por otro lado, en el caso de la enfermedad del Kuru, esta puede tardar hasta un año en el periodo de incubación y aún más en matar al paciente, de igual manera no se toman en cuenta partes como las características distintivas y únicas de cada sistema nervioso del cuerpo humano, lo cual afectaría de forma diferente a cada humano. Estos datos aumentarían la complejidad del algoritmo mientras más detallada sea la simulación.

Discusión.

En este apartado haremos algunas preguntas clave sobre el proyecto y sus resultados.

1.- ¿Es posible demostrar el comportamiento de un prion mediante un autómatas?

Sí, es posible modelar el comportamiento de un prion mediante la ocupación de autómatas. Así como los ejemplos dados en el proyecto siendo sencillos y muy generalizados, es posible detallarlos aún más si se hace una investigación exhaustiva sobre el sistema en donde se desarrollará el prion y la enfermedad.

2.- ¿La transportación del autómatas gráfico al codificado ha resultado clara y útil?

Sí, gracias a que los estados están bien definidos y sus transiciones son claras y explicadas, la transformación a un autómatas codificado es clara. Creamos funciones que nos definan los estados que la secuencia indique y, en el caso del primer autómatas, según sus condicionales sobre el inicio de la mutación del prion.

3.- ¿Es útil y aporta algo estas representaciones?

Sí, debido a que es importante conocer cómo es que estas infecciones actúan en el cuerpo de muchos animales, incluyendo a los humanos, el aporte de estos modelos del comportamiento de los priones y enfermedades como el Kuru es uno imprescindible, ya que se tiene otra herramienta de estudio para estos fenómenos y se pueden simular de forma general o detallada, según se necesite, por lo que éste y otros trabajos similares son de importancia para el campo.

Dado lo anterior, podemos determinar que se cumplió con el objetivo de explicar el comportamiento de un prion y una enfermedad causada por el mismo, ya que los autómatas y su codificación explican de forma clara la secuencia de los modelos y simulan las consecuencias que tiene el contagio de estos priones y los estragos que causa al cuerpo las enfermedades provocadas por los mismos.

Conclusión.

Los autómatas son herramientas matemáticas y computacionales extremadamente útiles para modelar una variedad de sistemas, incluyendo el comportamiento de enfermedades causadas por priones.

En el contexto de las enfermedades priónicas, los autómatas pueden ser utilizados para representar los diferentes estados que un paciente puede experimentar a lo largo del curso de la enfermedad. Cada estado puede representar una etapa diferente de la enfermedad, y las transiciones entre estados pueden representar los cambios que ocurren debido a la progresión de la enfermedad.

El uso de autómatas para modelar enfermedades priónicas tiene varias ventajas. En primer lugar, proporciona una representación visual y conceptual clara de la progresión de la enfermedad. Esto puede ser útil tanto para los investigadores que estudian la enfermedad como para los médicos que tratan a los pacientes.

En segundo lugar, los autómatas pueden ser utilizados para simular la progresión de la enfermedad bajo diferentes condiciones. Esto puede ayudar a los investigadores a entender mejor cómo se propaga la enfermedad y cómo podría ser controlada o tratada.

Finalmente, los autómatas pueden ser utilizados para predecir el curso de la enfermedad en un individuo o en una población. Esto puede ser útil para la planificación de la atención médica y para el desarrollo de estrategias de intervención.

Bibliografía.

IDE Online utilizado: www.onlinegdb.com/online_c_compiler

Kuru. (s/f). Medlineplus.gov. Recuperado el 13 de mayo de 2024, de <https://medlineplus.gov/spanish/ency/article/001379.htm>

LOS PRIONES Y SU BIOLOGÍA. (s/f). Svneurologia.org. Recuperado el 13 de mayo de 2024, de <http://www.svneurologia.org/congreso/priones-1.html>

Hiperactina, L. [@Lahiperactina]. (2020, agosto 2). Estas PROTEÍNAS son LETALES: ¿Qué son los PRIONES? Youtube. <https://www.youtube.com/watch?v=fp9-616Yfik>

Dai, J., Zhai, C., Ai, J., Ma, J., Wang, J., & Sun, W. (2020). Modeling the spread of epidemics based on cellular automata. Processes (Basel, Switzerland), 9(1), 55. <https://doi.org/10.3390/pr9010055>

(S/f). Europepmc.org. Recuperado el 13 de mayo de 2024, de <http://europepmc.org/article/pmc/2666960>