

UNIVERSIDAD DE SALAMANCA
MÁSTER UNIVERSITARIO EN MODELIZACIÓN MATEMÁTICA

Paradoja de Parrondo: estudio, avances y aplicaciones

AUTOR: Javier Palomino Sánchez

TUTORAS: María Dolores García Sanz

María Aurora Manrique García

Curso 2023-2024



VNiVERSiDAD
D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

UNIVERSIDAD DE SALAMANCA
MÁSTER UNIVERSITARIO EN MODELIZACIÓN MATEMÁTICA

Paradoja de Parrondo: estudio, avances y aplicaciones

AUTOR: Javier Palomino Sánchez

TUTORAS: María Dolores García Sanz

María Aurora Manrique García

Curso 2023-2024



MÁSTER UNIVERSITARIO EN MODELIZACIÓN MATEMÁTICA
Facultad de Ciencias
Universidad de Salamanca
e-mail: mumoma@usal.es
<https://www.usal.es/modelizacion-matematica>

CERTIFICADO DE LOS/AS TUTORES/AS DE TRABAJO DE FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN MODELIZACIÓN MATEMÁTICA

D^a María Dolores García Sanz y D^a María Aurora Manrique García, profesoras del Departamento de Economía e Historia Económica de la Universidad de Salamanca

HACEN CONSTAR

Que el trabajo titulado "Paradoja de Parrondo: estudio, avances y aplicaciones", que se presenta, ha sido realizado por Javier Palomino Sánchez, con DNI número ***34101M y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo Fin de Máster del Máster Universitario en Modelización Matemática de la Universidad de Salamanca

Salamanca, a fecha de la firma electrónica

Firmado

María Dolores García Sanz

María Aurora Manrique García

Índice general

Introducción	1
1 Preliminares	3
2 La paradoja de Parrondo	8
2.1 ¿Qué es el movimiento browniano?	8
2.2 Explicando la paradoja de Parrondo	11
3 Juegos dependientes de la historia y juegos cooperativos	28
3.1 Juegos dependientes de la historia	28
3.2 Juegos cooperativos	33
3.2.1 Primera variante	33
3.2.2 Segunda variante	41
4 Nuevos enfoques	44
4.1 Estudio del parámetro γ	44
4.2 Estudio del espacio de parámetros	47
4.3 ¿Existe la paradoja de Parrondo sobre tres juegos?	50
5 Aplicaciones	55
5.1 Aplicación sobre redes complejas	55
5.2 Estudio de situaciones síncronas	59
5.3 Otras aplicaciones	62
6 Conclusiones	64
7 Anexos	66
7.1 Anexo 1: Códigos de <i>Python</i>	66
7.2 Anexo 2: Códigos de <i>Mathematica</i>	99
Bibliografía	162

Índice de figuras

2.1	Estado encendido y apagado del potencial en la <i>flashing ratchet</i>	9
2.2	Potencial asimétrico de dientes de sierra	10
2.3	Reglas de los juegos A y B	12
2.4	Cadena de Markov para el juego A	14
2.5	Cadena de Markov para el juego B	15
2.6	Cadena de Markov reducida para el juego B	16
2.7	Evolución del capital promedio para el juego A	22
2.8	Evolución del capital promedio para el juego B	22
2.9	Evolución del capital promedio para la mezcla aleatoria	22
2.10	Comparación del capital promedio en la mezcla aleatoria en función del capital inicial	23
2.11	Comparación del capital promedio esperado en función de a y b	24
2.12	Evolución del capital según la secuencia ABBABB...	26
2.13	Comparación de la distribución de las monedas y el potencial de la <i>flashing ratchet</i>	27
3.1	Diagrama de árbol de los juegos dependientes de la historia	29
3.2	Cadena de Markov del juego dependiente de la historia B'	30
3.3	Cadena de Markov del juego B' reducida	30
3.4	Diagrama de árbol del juego cooperativo	34
3.5	Cadena de Markov del juego B''	38
3.6	Cadena de Markov del juego A modificado	42
4.1	Relación entre los parámetros γ y ϵ para los juegos originales	45
4.2	Relación entre los parámetros γ y ϵ para el juego dependiente de la historia	46
4.3	Relación entre los parámetros γ y ϵ para el juego cooperativo	46
4.4	Relación entre la probabilidad de ganar de la combinación aleatoria de los juegos originales A y B y el parámetro γ	47
4.5	Región donde el juego B es perdedor	48
4.6	Región donde los juegos A y B son perdedores	49
4.7	Región donde se da la paradoja de Parrondo	49
4.8	Cadena de Markov del juego A al considerar la mezcla aleatoria de A , B y C	51
5.1	Dinámicas del juego sobre redes complejas	56
5.2	Evolución del capital para la red completa con el patrón 4	57
5.3	Evolución del capital para la red de mundo pequeño con el patrón 4	58
5.4	Evolución del capital para la red de mundo pequeño separando en jugadores altruistas y egoístas	59

5.5	Evolución del capital para la situación democrática	60
5.6	Evolución del capital para la situación aleatoria	61
5.7	Evolución del capital para la situación del dictador	62

Índice de tablas

3.1	Condiciones de a , b y d para que se dé la paradoja en los juegos cooperativos verificando que B'' es justo para $\epsilon = 0$	40
-----	--	----

Introducción

La paradoja de Parrondo es un resultado aparentemente contradictorio perteneciente al campo de la teoría de juegos. La misma afirma que la combinación aleatoria o determinística de dos juegos perdedores puede dar lugar a un juego ganador. Su origen proviene del descubrimiento del efecto *ratchet* y su posterior aplicación por J. M. R. Parrondo al caso discreto tanto espacial como temporalmente. Una referencia muy recomendable para iniciarse en este tema es [33], un artículo introductorio redactado por el propio Parrondo en la Gaceta de la Real Sociedad Matematica Española.

El objetivo de este trabajo es explorar en profundidad esta paradoja, explicando los principales resultados que rodean al tema y comprobándolos de manera teórica y numérica, además de proponer nuevos resultados y avances y hacer una breve revisión sobre algunas de las aplicaciones que el tema tiene en campos tan diversos como las redes complejas, la toma de decisiones colectiva, la economía, la ecología o la física.

Para lograr estos objetivos, se ha seguido la siguiente metodología: se ha hecho una revisión bibliográfica del estado del tema desde sus inicios [2] hasta algunos de los artículos más recientes, como [5] y [29]. Además, también se han realizado simulaciones numéricas en *Python* [40] y *Mathematica* [48], que nos permiten comprobar los principales resultados asociados a la paradoja. A través de estas simulaciones hemos conseguido demostrar algunos nuevos resultados relacionados con este fenómeno.

El trabajo comienza con unos preliminares donde se muestran algunas de las nociones teóricas necesarias para trabajar posteriormente con las cadenas de Markov. Para la redacción de esta parte, nos hemos basado en [43]¹. Algunas otras nociones teóricas relativas a la modelización de procesos estocásticos empleadas en este capítulo 1 han sido extraídas de [46].

En el capítulo 2, se da un contexto histórico del descubrimiento de la paradoja y se profundiza más en ella. En esta parte encontramos algunas nociones físicas importantes y para su redacción nos hemos basado principalmente en [12], aunque también en [1], [7] y [15]. Para la explicación en profundidad de la paradoja hemos encontrado muy útil [33], aunque también se ha hecho uso de [18] y [47], sin olvidar otras referencias importantes como [16], [20], [34], [39] y [41]. También se han empleado otras referencias electrónicas como [3] y [32].

El capítulo 3 se dedica a la paradoja de Parrondo en dos tipos de juegos: los juegos dependientes de la historia y los juegos cooperativos. En la parte relativa a los juegos dependientes de la historia se ha hecho uso de [17], [19], [22], [23] y [38], mientras que

¹Apuntes de la profesora María Jesús Rivas López, de la asignatura “Modelos probabilísticos y estadísticos” impartida durante el curso académico 2023/2024 en el Máster Universitario en Modelización Matemática de la Universidad de Salamanca.

para la parte de los juegos cooperativos hemos revisado mayoritariamente [14] y [30], aunque también se han empleado los artículos [5], [31] y [49].

Para trabajar con la paradoja es importante hacer ver que nos alejamos de la noción clásica de juego, en la que se considera como tal una situación donde hay varios jugadores y las estrategias de cada uno afectan al resultado de los demás. Sin embargo, este tipo de juegos, denominados juegos de Parrondo, se juegan, en el modelo inicial, por un único jugador y no se eligen estrategias en ellos, como iremos viendo en el trabajo. En el caso de los juegos cooperativos sí aparecen varios jugadores, pero no se consideran coaliciones entre ellos.

En los juegos dependientes de la historia, la probabilidad con la que el jugador gana el juego depende del resultado en sus dos turnos anteriores. En los juegos cooperativos, consideraremos varios jugadores y la probabilidad de ganar de cada jugador dependerá del resultado de sus vecinos a izquierda y derecha.

En el capítulo 4 exponemos algunos nuevos resultados a los que hemos llegado. Para ello, se han realizado generalizaciones sobre algunos resultados ya conocidos y también se han investigado otros nuevos basados en ideas propias o propuestas en artículos estudiados, que han dado lugar a nuevos avances en el tema.

El capítulo 5 está dedicado a la aplicación de la paradoja en redes complejas y en situaciones síncronas. Además, se han incluido algunas aplicaciones de la paradoja en otros campos científicos. Una fuerte inspiración para esta parte ha sido [27], aunque también se ha trabajado con los artículos [10], [25], [29], [35], [36] y [37].

Finalmente, el capítulo 6 incluye las conclusiones alcanzadas en el trabajo. En el capítulo 7 se encuentran los anexos 7.1 y 7.2, que recogen respectivamente los códigos de *Python* y de *Mathematica* empleados para las simulaciones numéricas realizadas. Ninguno de estos códigos es estrictamente necesario para entender el trabajo, pero sirven de apoyo para el mismo y confío en que el lector interesado en el tema los encontrará útiles.

Capítulo 1

Preliminares

Sea Ω un espacio muestral asociado a un experimento aleatorio, A la σ -álgebra de sucesos y $\mathbb{P}$ una función de probabilidad $\mathbb{P} : (\Omega, A) \rightarrow [0, 1]$ sobre A .

Definición 1.0.1. Una **variable aleatoria** sobre Ω es una función $X : (\Omega, A) \rightarrow \mathbb{R}$ tal que la antiimagen de cualquier intervalo real pertenece a la σ -álgebra A .

Observación 1.0.1. Denotamos por $\mathbb{P}_X$ a la probabilidad inducida por la variable X en $\mathbb{R}$.

$$\begin{array}{ccc} (\Omega, A) & \xrightarrow{X} & \mathbb{R} \\ & \searrow \mathbb{P} & \downarrow \mathbb{P}_X \\ & & [0, 1] \end{array}$$

Definición 1.0.2. Un **proceso estocástico** es una aplicación

$$\begin{aligned} X : \Omega \times T &\longrightarrow E \\ (\omega, t) &\longrightarrow X(\omega, t) = X_t(\omega) = e \end{aligned}$$

donde $E \subseteq \mathbb{R}$ y T es el conjunto de índices temporales (en procesos de tiempos discretos T es el conjunto de los números naturales, aunque en procesos de tiempo continuo puede ser el conjunto de números reales no negativos), de manera que:

- Fijado $\omega \in \Omega$, $X(\omega, \cdot) : T \rightarrow E$ es la función trayectoria de la partícula ω .
- Fijado $t \in T$, $X(\cdot, t) = X_t : \Omega \rightarrow E$ es una variable aleatoria.

Para cada tiempo t existirá una probabilidad $\mathbb{P}_{X_t}$ inducida por la variable X_t .

$$\begin{array}{ccc} \Omega & \xrightarrow{X_t} & \mathbb{R} \\ & \searrow \mathbb{P} & \downarrow \mathbb{P}_{X_t} \\ & & [0, 1] \end{array}$$

Observación 1.0.2. Un proceso estocástico puede verse como la evolución a lo largo del tiempo de una variable aleatoria. Esta familia uniparamétrica de variables aleatorias permite modelizar procesos dinámicos en los que existe cierta aleatoriedad.

Observación 1.0.3. En este contexto se suele denominar **espacio de partículas** a Ω , **espacio de tiempos** a T y **espacio de estados** a E .

Observación 1.0.4. Si $X(\omega, t) = X_t(\omega) = e$, diremos que la partícula ω en el instante t está en el estado e .

Definición 1.0.3. Una **cadena** es un proceso estocástico con espacio de tiempos discreto $T = \{t_0, t_1, t_2, \dots\}$, y espacio de estados finito o numerable $E = \{e_0, e_1, e_2, \dots\}$ en el que una variable aleatoria $X_{t_n} = X_n$ va cambiando con el tiempo.

Definición 1.0.4. Diremos que una cadena es una **cadena de Markov** cuando verifica la **propiedad de Markov**: la distribución de probabilidad de su valor futuro depende únicamente de su valor presente, siendo independiente de su historia pasada. Matemáticamente, se tiene que:

$$\mathbb{P}(X_{n+1} = e_{n+1} \mid X_0 = e_0, X_1 = e_1, \dots, X_n = e_n) = \mathbb{P}(X_{n+1} = e_{n+1} \mid X_n = e_n)$$

Definición 1.0.5. Para cualquier etapa n (es decir, un instante $t_n \in T$) y para cualesquiera $e_i, e_j \in E$, las probabilidades condicionadas $p_{ij}(n) = \mathbb{P}(X_n = e_j \mid X_{n-1} = e_i)$ se denominan **probabilidades de transición** (en una etapa) del estado e_i al estado e_j .

Observación 1.0.5. Las probabilidades de transición satisfacen $p_{ij}(n) \geq 0$ y $\sum_{e_j \in E} p_{ij}(n) = 1$ para cada $e_i \in E$ y dependen de la etapa n escogida.

Definición 1.0.6. Diremos que una matriz es **estocástica** cuando es una matriz cuadrada de elementos no negativos en la que la suma de los elementos de cada fila es igual a 1.

Definición 1.0.7. Diremos que una cadena es **homogénea**, o que tiene **probabilidades de transición estacionarias**, cuando sus probabilidades de transición no dependen de la etapa, esto es:

$$p_{ij}(n) = \mathbb{P}(X_n = e_j \mid X_{n-1} = e_i) = \mathbb{P}(X_m = e_j \mid X_{m-1} = e_i) = p_{ij}(m) \quad \forall n, m \in \mathbb{N}, \forall e_i, e_j \in E$$

Observación 1.0.6. En este caso, denotaremos por p_{ij} a la probabilidad de que, al pasar de una etapa a la siguiente, la cadena pase del estado e_i al e_j . Así, estas probabilidades verifican:

$$p_{ij} = \mathbb{P}(X_n = e_j \mid X_{n-1} = e_i), \quad p_{ij} \geq 0, \quad \sum_{e_j \in E} p_{ij} = 1 \quad \forall n \in \mathbb{N}, \forall e_i, e_j \in E$$

Definición 1.0.8. Se denomina **matriz de transición** (en una etapa) a la matriz cuadrada de orden $m = \text{card}(E)$ formada por las probabilidades de transición, de forma que los subíndices de las filas indican estados de partida y los de las columnas estados de llegada.

$$\mathcal{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{pmatrix}$$

Observación 1.0.7. La matriz de transición es estocástica.

Observación 1.0.8. Como el producto de matrices estocásticas es otra matriz estocástica, cualquier potencia de la matriz de transición es también estocástica.

Definición 1.0.9. Dada una cadena homogénea, se define la **probabilidad de transición desde el estado e_i al estado e_j en n etapas** como la probabilidad de que la cadena esté en el estado e_j si n etapas antes estaba en el estado e_i .

$$p_{ij}^{(n)} = \mathbb{P}(X_n = e_j \mid X_0 = e_i)$$

Con estas probabilidades se puede construir la **matriz de transición en n etapas**, que denotamos $\mathcal{P}^{(n)}$ y que coincide con la potencia n -ésima de la matriz de transición, es decir, $\mathcal{P}^{(n)} = \mathcal{P}^n$.

Definición 1.0.10. Se denomina **distribución inicial** de una cadena al vector de probabilidades $d^{(0)} = (d_1^{(0)}, d_2^{(0)}, \dots, d_m^{(0)})$, donde $d_i^{(0)} = \mathbb{P}(X_0 = e_i)$, $\forall e_i \in E$.

Observación 1.0.9. Al ser la distribución inicial una distribución de probabilidad, se verifica $d_i^{(0)} \geq 0 \forall e_i \in E$ y $\sum_{e_i \in E} d_i^{(0)} = 1$.

Definición 1.0.11. La **distribución marginal de la variable X_n** es el vector de probabilidades $d^{(n)} = (d_1^{(n)}, d_2^{(n)}, \dots, d_m^{(n)})$, donde $d_i^{(n)} = \mathbb{P}(X_n = e_i)$, $\forall e_i \in E$.

Observación 1.0.10. Se verifica que $d_i^{(n)} \geq 0 \forall e_i \in E$ y $\sum_{e_i \in E} d_i^{(n)} = 1$. Este vector indica la probabilidad de que la cadena esté en un estado $e_i \in E$ en el instante n (o después de n etapas).

Observación 1.0.11. Se tiene que $d^{(m+n)} = d^{(m)}\mathcal{P}^n$, $\forall m, n \in \mathbb{N}$. En particular, se verifica que $d^{(n)} = d^{(0)}\mathcal{P}^n$, $\forall n \in \mathbb{N}$.

Observación 1.0.12. En cadenas de Markov homogéneas, la matriz de transición y la distribución inicial bastan para determinar cualquier otra probabilidad asociada a la cadena.

Definición 1.0.12. Atendiendo al número de estados que una cadena de Markov puede alcanzar, las cadenas se clasifican en **infinitas** (cuando $\text{card}(E) = \infty$) y **finitas** (cuando $\text{card}(E) < \infty$).

Definición 1.0.13. El estado e_i **comunica** con el estado e_j ($e_i \rightarrow e_j$) si hay probabilidad positiva de que la cadena llegue al estado e_j partiendo del estado e_i , en algún número de etapas; es decir, $e_i \rightarrow e_j$ si y solo si $\exists n \in \mathbb{N}$ tal que $p_{ij}^{(n)} > 0$. Dos estados e_i y e_j **intercomunican** ($e_i \leftrightarrow e_j$) cuando el estado e_i comunica con el e_j y el e_j comunica con el e_i ; es decir, $e_i \leftrightarrow e_j$ si y solo si $\exists n, m \in \mathbb{N}$ tales que $p_{ij}^{(n)}, p_{ji}^{(m)} > 0$.

Definición 1.0.14. Decimos que un conjunto de estados $C \subseteq E$ es **irreducible** cuando cualquier par de estados de C intercomunican. Cuando el conjunto de estados E es irreducible se dice que la cadena es **irreducible**.

Definición 1.0.15. Un estado e_i se dice que es **transitorio** cuando si la cadena sale de e_i no es seguro que vuelva a pasar por e_i .

$$P(X_n = e_i \text{ para algún } n \geq 1 \mid X_0 = e_i) < 1$$

Definición 1.0.16. Un estado e_i es **recurrente** cuando si la cadena sale de e_i es seguro que vuelve a pasar por e_i .

$$\mathbb{P}(X_n = e_i \text{ para algún } n \geq 1 \mid X_0 = e_i) = 1$$

Definición 1.0.17. Dada una cadena de Markov con un conjunto de estados E podemos asociar a cada estado $e_i \in E$ una variable aleatoria T_i , que es el número de pasos dados en un camino aleatorio en la cadena que empieza y acaba en e_i . De esta manera, el **tiempo medio de recurrencia** μ_i de un estado $e_i \in E$ se define como $\mu_i = \mathbb{E}[T_i]$.

Definición 1.0.18. Un estado recurrente e_i se dice que es **recurrente nulo** si su tiempo medio de recurrencia $\mu_i = \infty$ (solo en cadenas infinitas) y **recurrente positivo** si $\mu_i < \infty$.

Definición 1.0.19. Un estado recurrente e_i es **periódico** si el número de etapas que tienen que pasar entre la salida y el regreso de la cadena a dicho estado es múltiplo de $t \geq 2$, siendo $t = \text{m.c.d.}\{n \mid p_{ii}^{(n)} > 0\}$; es decir, $p_{ii}^{(n)} \neq 0$ para $n = t, 2t, 3t, \dots$. Es **aperiódico** si es recurrente y $t = 1$.

Definición 1.0.20. Un estado es **ergódico** si es un estado recurrente positivo y aperiódico.

Definición 1.0.21. Una cadena de Markov **ergódica** es una cadena irreducible de estados ergódicos; es decir, todos sus estados intercomunican y son recurrentes positivos y aperiódicos.

Definición 1.0.22. Diremos que una distribución marginal es **estacionaria** cuando se mantiene inalterable, como distribución marginal, con la evolución temporal de la cadena. Es decir, dada una distribución $\pi = (\pi_1, \pi_2, \dots, \pi_m)$, es estacionaria cuando se verifica $\pi \cdot \mathcal{P} = \pi$.

Observación 1.0.13. Esta igualdad implica un sistema de $m + 1$ ecuaciones:

- m ecuaciones de equilibrio:

$$\pi_j = \sum_{e_i \in E} \pi_i p_{ij}, \quad j = 1, 2, \dots, m$$

- Una ecuación normalizadora:

$$\sum_{e_i \in E} \pi_i = 1, \quad \pi_i \geq 0$$

Observación 1.0.14. Si la distribución inicial $d^{(0)} = \pi$ es estacionaria se verifica que

$$d^{(n)} = \pi \mathcal{P}^n = \pi \mathcal{P} \mathcal{P}^{n-1} = \pi \mathcal{P} \mathcal{P}^{n-2} = \dots = \pi \mathcal{P} = \pi$$

Observación 1.0.15. Si π es estacionaria para $\mathcal{P}$, también lo es para $\mathcal{P}^n$.

Los teoremas 1.0.1 y 1.0.2 enunciados a continuación son fundamentales para los resultados de la paradoja de Parrondo que estudiaremos.

Teorema 1.0.1. *Si una cadena de Markov finita es ergódica entonces tiene una única distribución estacionaria $\pi = (\pi_1, \pi_2, \dots, \pi_m)$ que viene dada por $\pi_j = 1/\mu_j$ para todo $e_j \in E$ donde μ_j es el tiempo medio de recurrencia del estado e_j .*

Observación 1.0.16. El sistema de ecuaciones que se genera es un sistema compatible determinado de $m+1$ ecuaciones con m incógnitas. Por tanto, se puede eliminar una de las ecuaciones del sistema, teniendo en cuenta que no puede ser la ecuación de normalización.

Definición 1.0.23. Llamaremos **probabilidades de transición límite** al límite de las probabilidades de transición en n etapas cuando el número de etapas tiende a infinito, $p_{ij}^{(\infty)} = \lim_{n \rightarrow \infty} p_{ij}^{(n)}$.

Observación 1.0.17. Así mismo, si $\mathcal{P}$ es la matriz de transición de una cadena de Markov finita se puede estudiar cómo es la matriz de transición en n etapas, $\mathcal{P}^{(n)} = \mathcal{P}^n$ cuando el número de etapas va aumentando; así, si existe, la matriz de transición límite será $\mathcal{P}^\infty = \lim_{n \rightarrow \infty} \mathcal{P}^n$.

Definición 1.0.24. Denominamos **distribución límite** a la distribución de los estados a la que tiende la cadena cuando el número de etapas tiende a infinito, $d^{(\infty)} = d^{(0)} \cdot \mathcal{P}^\infty$.

Teorema 1.0.2. *Las probabilidades de transición límite de una cadena de Markov finita ergódica con matriz de transición $\mathcal{P}$ vienen dadas por*

$$p_{ij}^{(\infty)} = \lim_{n \rightarrow \infty} p_{ij}^{(n)} = \frac{1}{\mu_j}$$

para todo par de estados $e_i, e_j \in E$, siendo μ_j el tiempo medio de recurrencia del estado e_j .

Por tanto, la matriz de transición límite de una cadena de Markov ergódica es

$$\mathcal{P}^\infty = \lim_{n \rightarrow \infty} \mathcal{P}^n = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_m \\ \pi_1 & \pi_2 & \cdots & \pi_m \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_m \end{pmatrix}$$

Como la suma de las componentes de una distribución marginal es 1, se verifica

$$d^{(\infty)} = d^{(0)} \cdot \mathcal{P}^\infty = \left(\pi_1 \sum_{e_i \in E} d_i^{(0)}, \pi_2 \sum_{e_i \in E} d_i^{(0)}, \dots, \pi_m \sum_{e_i \in E} d_i^{(0)} \right) = (\pi_1, \pi_2, \dots, \pi_m) = \pi$$

Es decir, en una cadena de Markov finita y ergódica, la distribución límite coincide con la estacionaria.

Capítulo 2

La paradoja de Parrondo

2.1. ¿Qué es el movimiento browniano?

En el año 1827, el botánico Robert Brown (Montrose, 1773) descubrió un curioso fenómeno [6]. Mientras se encontraba trabajando con partículas de aproximadamente unas micras de diámetro que provenían de granos de polen triturados y que posteriormente habían sido disueltos en agua, el botánico observó, de manera microscópica, que las partículas presentaban un movimiento errático, pese a que tanto la muestra como el microscopio se encontraban en total estado de reposo.

Al principio, Brown conjeturó que este movimiento presente en las partículas era debido a su origen biológico. Sin embargo, posteriormente dicha idea fue descartada al hacer el mismo experimento con diferentes tipos de partículas y comprobar así que todas ellas se comportaban de esa manera, siempre y cuando tuvieran un tamaño lo suficientemente pequeño. Desde entonces, este movimiento errático presente en partículas pequeñas se conoce como *movimiento browniano*, en honor a su descubridor.

Más adelante, en el campo de la física, se empezó a conocer como *partícula browniana* a toda partícula que, en contacto con un baño térmico y debido a su pequeño tamaño y masa, realizase un movimiento irregular. Tratemos de entender un poco mejor la razón de este errático movimiento.

A finales del siglo XIX ya se propusieron las primeras teorías que argumentaban que el movimiento browniano era una consecuencia de la naturaleza molecular intrínseca de la materia. Para explicarlo, se argumentó que pese a que el fluido en el que las partículas brownianas están contenidas se encuentra en estado de equilibrio, las moléculas del mismo se mueven muy rápidamente a temperatura ambiente. De esta manera, algunas moléculas del fluido chocarán con la partícula browniana considerada, proveniendo estas de cualquier dirección.

Como el fluido está en reposo, el momento total promedio transferido por las moléculas del fluido a la partícula browniana ha de ser nulo. De esta manera, cuando la masa de la partícula sea lo suficientemente pequeña, cualquier mínima diferencia entre el momento transferido en una dirección con un determinado sentido y en dicha dirección con sentido contrario hará que la partícula comience a moverse de manera apreciable para el microscopio. Una nueva variación del momento transferido hará que la partícula cambie su movimiento, dando origen así al movimiento errático observado en el microscopio.

Ante el descubrimiento de las partículas brownianas, se empezaron a crear los llamados

motores brownianos, sistemas capaces de rectificar, por lo menos de manera parcial, el movimiento errático de estas partículas. Para ello, emplean algún tipo de asimetría espacial o temporal. Dicha asimetría transporta en una cierta dirección a la partícula, lo que permite controlar el movimiento browniano presente en ella.

Pese a que con el término de motor browniano se han dado a conocer una gran cantidad de sistemas y modelos diferentes, en el presente trabajo nos centramos principalmente en el motor browniano denominado *flashing ratchet*.

En este motor browniano se consideran una o varias partículas brownianas. A estas se les aplica un potencial asimétrico de dientes de sierra unidimensional. Dicho potencial puede ser encendido o apagado por un agente que lo controla. El potencial se representa en la figura 2.1:

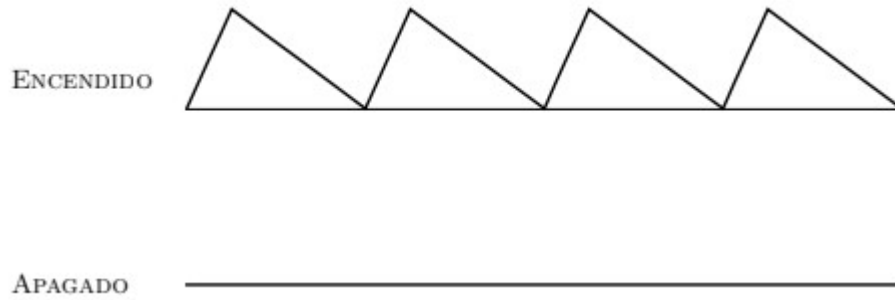


Figura 2.1: Estado encendido y apagado del potencial en la *flashing ratchet*. Figura extraída de [12].

En dicho gráfico, el eje horizontal representa la posición de la partícula y el eje vertical el potencial. Experimentalmente se observa que si se enciende y se apaga el motor browniano de manera periódica, las partículas se ven transportadas en el sentido de la más débil de las dos fuerzas generadas por el potencial, en el caso de la figura anterior, hacia la derecha. Más sorprendente aún resulta el hecho de que este fenómeno también sucede cuando el encendido y apagado del motor browniano se realiza de manera aleatoria. Este hecho, conocido como *efecto ratchet*, resulta imprevisible en un primer momento ya que, tanto si el potencial se enciende como si se apaga de manera permanente, las partículas alcanzan un estado de equilibrio y por lo tanto no se ven afectadas por ningún tipo de transporte.

De esta manera, la *flashing ratchet* consiste en una partícula browniana en una dimensión sometida a un potencial, siendo este periódico en el espacio pero asimétrico. Dicho potencial puede ser conectado o desconectado según la decisión de un controlador externo. Siguiendo a [12] para entenderlo mejor podemos describir la dinámica de la partícula browniana mediante la siguiente ecuación de Langevin:

$$\gamma \dot{X}(t) = -\alpha(t)\dot{V}(x) + \xi(t)$$

donde $X(t)$ representa la posición de la partícula en el instante t , $\dot{X} = \frac{dX}{dt}$ es la derivada temporal de la variable $X(t)$, γ es el coeficiente de fricción de la partícula, y $\xi(t)$ es un ruido blanco gaussiano.

El potencial $V(x)$ presenta periodo L y es asimétrico, de modo que $-\dot{V}(x)$ es la fuerza que siente una partícula colocada en la posición x debido al potencial. En el caso del *flashing ratchet*, el potencial $V(x)$ está formado por dos tramos rectos de distinta pendiente, y se define como

$$V(x) = \begin{cases} \frac{V_0}{a} \frac{x}{L} & \text{si } 0 \leq \frac{x}{L} \leq a \\ -\frac{V_0}{1-a} \left(\frac{x}{L} - a \right) + V_0 & \text{si } a \leq \frac{x}{L} \leq 1 \end{cases}$$

Además, se extiende de manera periódica $V(x + L) = V(x)$ más allá de la región $[0, L]$ como se ve en la figura 2.2:

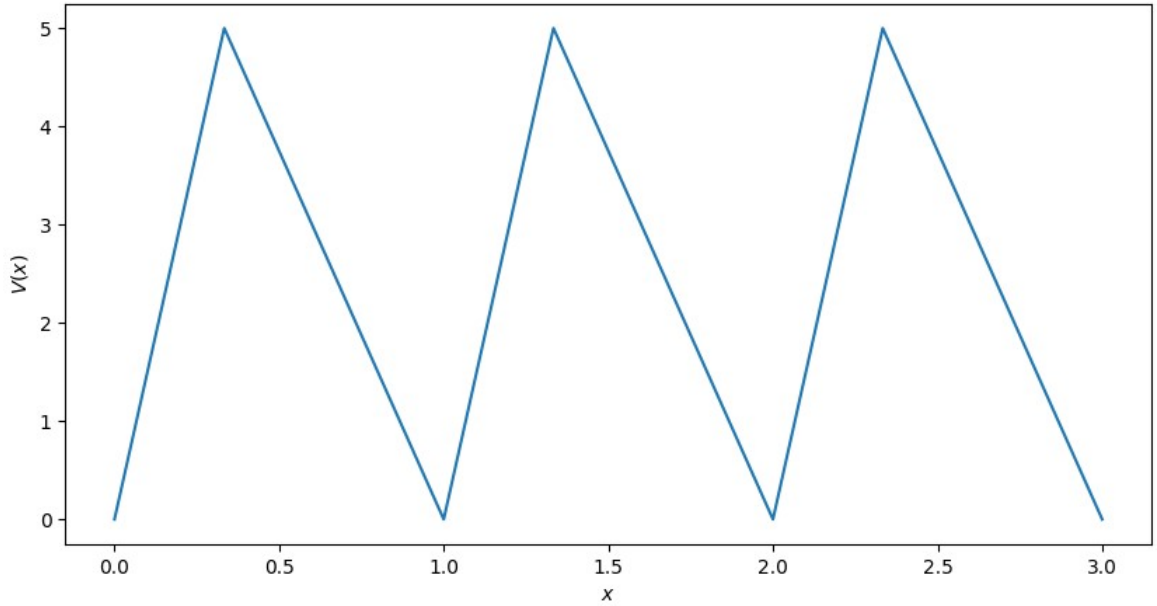


Figura 2.2: Potencial asimétrico de dientes de sierra. Elaboración propia.

Para elaborar el gráfico anterior, se han empleado los valores $L = 1$, $V_0 = 5$ y $a = 1/3$ y para construirlo se ha empleado el código de *Python* presente en el anexo 7.1.1.

Por construcción, la asimetría viene controlada por el parámetro $0 < a < 1$, de manera que el potencial es simétrico cuando a toma el valor $1/2$ y asimétrico en caso contrario. De igual manera, el parámetro V_0 controla la intensidad del potencial.

Se denomina “tramo corto” a la región del eje x donde el potencial tiene pendiente positiva, esto es, la que viene definida por $0 \leq x/L \leq a$, y “tramo largo” al tramo donde tiene pendiente negativa.

Finalmente, $\alpha(t)$ sólo puede tomar el valor 0 o 1, de modo que controla si el potencial está encendido o apagado, ya que toma el valor $\alpha = 1$ cuando el potencial se encuentra encendido y $\alpha = 0$ en caso contrario.

De esta manera, el efecto *ratchet* demuestra que, en general, la alternancia entre dos dinámicas aleatorias no tiene por qué ser una combinación de los efectos de cada una sino que puede dar lugar a una dinámica completamente inesperada. Este sorprendente hecho lo relacionaremos posteriormente con los juegos de azar ya que, como veremos, dos juegos justos, en los que el jugador ni gana ni pierde capital en promedio, o incluso dos juegos perdedores, pueden dar lugar a un juego ganador al combinarlos de manera periódica e

incluso aleatoriamente. Este traslado del fenómeno del movimiento browniano presente en partículas físicas al mundo de los juegos de azar se denominó *paradoja de Parrondo*, a la que dedicamos la siguiente sección.

2.2. Explicando la paradoja de Parrondo

Antes de hablar de la paradoja de Parrondo, debemos entender a qué nos referimos con el término “paradoja”. Según la RAE, se entiende por paradoja un “hecho o dicho aparentemente contrario a la lógica”. Nos atañeremos a esta acepción de paradoja en este trabajo.

Una vez entendido el término, la paradoja de Parrondo se definió como un resultado aparentemente ilógico o paradójico en relación a la teoría de juegos. Ahondemos un poco en ella.

Juan Manuel Rodríguez Parrondo (Madrid, 1964) es un físico español dedicado al campo de la física estadística. Investigando en ese campo y específicamente en el concepto del movimiento browniano, en 1996 Parrondo demostró que la rectificación de fluctuaciones también es cierta cuando el tiempo y el grado de libertad de la partícula browniana empleada son discretos.

De esta manera, Parrondo reinterpretó la posición unidimensional y discreta de la partícula como el capital de un jugador que juega a un determinado juego, y la coordenada temporal como el número de turnos del juego. Con todo esto, formuló una versión discreta de la *flashing ratchet* en términos de dos juegos de azar. En ella, se presentan dos juegos, A y B , que pueden ser justos o perdedores (en el sentido de que el capital esperado del jugador es o bien una función constante con el paso del tiempo o bien decreciente). Combinando estos dos juegos, ya sea de una manera periódica o incluso completamente aleatoria, se genera un juego ganador, que hace que el capital esperado del jugador sea una función creciente. Estos dos juegos, que procederemos a explicar posteriormente, son la base de lo que se ha dado en llamar desde entonces la paradoja de Parrondo.

La paradoja fue nombrada por primera vez como tal en 1999, tras una publicación de los ingenieros Derek Abbott y Greg P. Harmer [2]. Derek Abbott había coincidido con Parrondo durante una visita de este a la universidad de Adelaida, en Australia, donde se inició en el tema.

Procedamos ahora a explicar de una manera más técnica qué es la paradoja de Parrondo. Como mencionamos en la introducción, los juegos empleados aquí no coinciden con la definición clásica de juego, ya que son jugados en principio por un único jugador y no se consideran estrategias en ellos. Consideremos dos juegos de este tipo, en principio de un solo jugador, que denominaremos A y B . En cada uno de los turnos del juego, el jugador apuesta una unidad de capital, por ejemplo podríamos pensar en un euro. De esta manera, denotaremos como $X(t)$ al capital acumulado por el jugador en cada una de las réplicas o turnos del juego, $t = 0, 1, 2, \dots$

El juego A consiste en el lanzamiento de una moneda ligeramente sesgada hacia una de sus caras, de modo que la probabilidad de que el jugador gane es algo menor que $1/2$. Este sesgo en la moneda puede ser controlado mediante un parámetro ϵ , haciendo $p_A = \frac{1}{2} - \epsilon$, donde ϵ es un número pequeño y positivo, y p_A es la probabilidad de ganar el juego A , con lo que $1 - p_A = \frac{1}{2} + \epsilon$ es la probabilidad de perderlo, es decir, de que el jugador pierda

una unidad de capital.

Observación 2.2.1. De manera general, no es importante el capital inicial con el que el jugador empieza a jugar, ya que lo único que nos interesa es si este crece o decrece. De hecho, se podrían dar incluso casos en los que el capital del jugador fuera negativo, pues esta situación sería equivalente a empezar a jugar con un capital lo suficientemente grande como para que este no llegue a cero.

Para explicar el juego B , consideramos dos monedas sesgadas. A una de estas monedas la etiquetaremos como la moneda “buena”, siendo la otra moneda la moneda “mala”. En cada turno t del juego B se observa primeramente el capital del jugador, $X(t)$. Si $X(t)$ es múltiplo de 3, el jugador debe lanzar la moneda mala. Con esta moneda, la probabilidad de ganar el juego es de tan solo $p_m = \frac{1}{10} - \epsilon$. En cualquier otro caso, es decir, cuando el capital del jugador no es múltiplo de 3, se lanza la moneda buena y la probabilidad de ganar es $p_b = \frac{3}{4} - \epsilon$.

Modelizado el juego de esta manera, en los turnos en los que el capital del jugador no sea un múltiplo de 3, el resultado más probable es que el jugador gane el juego y con ello su capital aumente en una unidad, mientras que cuando su capital sea múltiplo de 3 es casi seguro que el jugador perderá el juego y con ello tendrá una unidad de capital menos.

Observación 2.2.2. Al emplear esta parametrización es necesario que $\epsilon \in [0, \frac{1}{10}]$ para que las probabilidades tomen valores entre 0 y 1.

Las reglas que muestran las dinámicas de ambos juegos se sintetizan en la figura 2.3, en la que los tonos representan las probabilidades de ganar de cada moneda, siendo mayor la probabilidad de ganar cuanto más claro sea el tono:

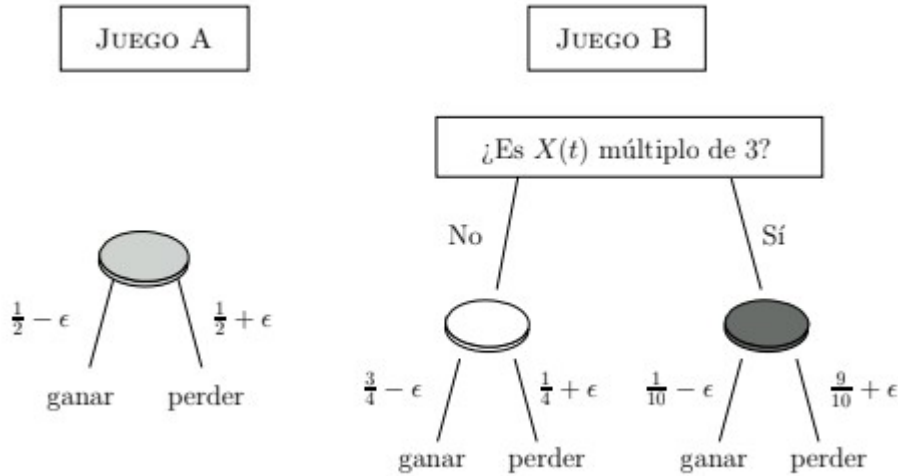


Figura 2.3: Reglas de los juegos A y B . Figura extraída de [12].

Observación 2.2.3. En la paradoja de Parrondo original, se tomó como $M = 3$ el número del cual el capital debe ser múltiplo para que tiremos la moneda mala. Sin embargo, nada nos impide generalizar el juego a cualquier otro valor natural de M , aunque en los cálculos en este trabajo se ha optado por respetar la decisión de Parrondo, salvo que se indique lo contrario.

Observación 2.2.4. Entendiendo el capital del jugador como una variable aleatoria que evoluciona con el número de turnos, esto es, un proceso estocástico, diremos que un juego es **ganador** cuando el capital promedio, que va evolucionando según se juega al mismo, sea una función creciente. De la misma manera, diremos que un juego es **perdedor** si el capital promedio decrece con el número de turnos. Si no sucede ni lo uno ni lo otro, sino que es constante, consideraremos que el juego es **justo**.

Observación 2.2.5. Como veremos más adelante, en ocasiones también puede ser útil considerar la ganancia esperada, es decir, la esperanza de la variable aleatoria dada por la diferencia de las variables aleatorias $X(t+1)$ y $X(t)$, esto es, $\mathbb{E}(X(t+1) - X(t))$. De esta manera, estaremos ante un juego ganador cuando la ganancia esperada sea positiva, perdedor cuando sea negativa y justo cuando sea nula.

Con esta definición se tiene que tanto el juego A como el B son justos si $\epsilon = 0$ y perdedores para $\epsilon > 0$. Que el juego A es justo cuando $\epsilon = 0$ es algo trivial, puesto que la probabilidad de ganar o de perder el juego es la misma en este caso.

En el caso del juego B , la demostración de este resultado es algo más compleja y se necesitan conceptos de cadenas de Markov, ya que la moneda empleada depende a su vez del capital con el que contemos. Lo veremos de manera detallada más adelante.

En el caso del juego A , la variable aleatoria que mide el capital del jugador en cada turno t , es la siguiente:

$$X(t+1) = \begin{cases} X(t) + 1 & \text{con probabilidad } p \\ X(t) - 1 & \text{con probabilidad } 1 - p \end{cases}$$

Como se mencionó anteriormente, nos será más fácil trabajar con la diferencia $Z(t) = X(t+1) - X(t)$, esto es:

$$Z(t) = X(t+1) - X(t) = \begin{cases} 1 & \text{con probabilidad } p \\ -1 & \text{con probabilidad } 1 - p \end{cases}$$

La cantidad promedio en la que crece el capital en un turno para el juego A , esto es, la ganancia del juego A , será:

$$\mathbb{E}(Z(t)) = \mathbb{E}(X(t+1) - X(t)) = (+1)p + (-1)(1 - p) = 2p - 1$$

Confirmamos entonces que si la probabilidad de ganar es $p = 1/2$, el juego A será un juego justo según nuestra definición anterior, pues la ganancia esperada sería nula. De manera análoga, si la probabilidad de ganar es mayor que $1/2$ estaremos ante un juego ganador y en caso contrario ante un juego perdedor.

En vez de calcular la esperanza de la diferencia $X(t+1) - X(t)$, podemos calcular la esperanza de la propia variable aleatoria $X(t)$. Como se tiene que $\mathbb{E}(X(t+1) - X(t)) = 2p - 1$, entonces tenemos que $\mathbb{E}(X(t)) = t(2p - 1)$, que no es más que la ganancia en cada turno multiplicada por el número de turnos, y donde estamos suponiendo que el capital inicial era 0, $X(0) = 0$.

Siguiendo un razonamiento análogo, podemos calcular la varianza de $Z(t)$:

$$\text{Var}(Z(t)) = \mathbb{E}(Z(t)^2) - \mathbb{E}(Z(t))^2$$

Como

$$Z(t)^2 = \begin{cases} 1^2 = 1 & \text{con probabilidad } p \\ (-1)^2 = 1 & \text{con probabilidad } 1 - p \end{cases}$$

se tiene que

$$Z(t)^2 = 1 \text{ con probabilidad } 1$$

y por lo tanto

$$\mathbb{E}(Z(t)^2) = 1$$

Además, como

$$\mathbb{E}(Z(t))^2 = (2p - 1)^2 = 4p^2 + 1 - 4p$$

se obtiene que

$$\text{Var}(Z(t)) = \mathbb{E}(Z(t)^2) - \mathbb{E}(Z(t))^2 = 1 - (4p^2 + 1 - 4p) = 4p - 4p^2 = 4p(1 - p)$$

Como la varianza de $Z(t)$ es $4p(1 - p)$, la varianza de $X(t)$ es la multiplicación de esta por el número de turnos, es decir:

$$\text{Var}(X(t)) = 4p(1 - p)t$$

Por lo tanto, el capital $X(t)$ jugando siempre al juego A tiene una esperanza de $t(2p - 1)$ y una varianza de $4p(1 - p)t$, donde p es la probabilidad de ganar A .

Ahora podemos realizar un análisis del juego utilizando la teoría de cadenas de Markov. Tal y como está modelizado el juego A , es fácil ver que la cadena de Markov asociada a este juego es la dada en la figura 2.4:

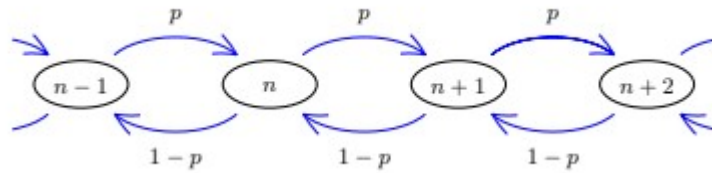


Figura 2.4: Cadena de Markov para el juego A . Figura extraída de [18].

De esta manera estamos ante una cadena infinita donde cada uno de los estados representa el capital del jugador. Desde un estado pasamos al estado siguiente con probabilidad p y retrocedemos hacia el estado anterior con probabilidad $1 - p$.

La matriz de transición asociada a esta cadena de Markov, cuyos elementos son todos nulos salvo en las dos diagonales por debajo y por encima de la diagonal principal, es la siguiente:

$$\mathcal{P}_A = \begin{pmatrix} 0 & p & 0 & 0 & (1-p) \\ 1-p & 0 & \ddots & 0 & 0 \\ 0 & 1-p & \ddots & p & 0 \\ 0 & 0 & \ddots & 0 & \ddots \\ (p) & 0 & 0 & \ddots & \ddots \end{pmatrix}$$

En la práctica, tanto la cadena de Markov del juego A como su matriz asociada no tienen por qué ser infinitas, sino que las podemos restringir al rango de interés. Por ejemplo, si se hacen n jugadas del juego A valdrá con una cadena de $2n + 1$ estados, y por lo tanto bastará con una matriz cuadrada de orden $2n + 1$. De esta manera, los dos términos que aparecen entre paréntesis en la matriz $\mathcal{P}_A$ simplemente se han puesto por completitud, pues nunca se llegarán a necesitar.

Para trabajar con el juego B es necesario realizar un análisis algo más complejo ya que la probabilidad de ganar en cada turno depende del capital en dicho turno. Sin embargo, veremos que para $\epsilon = 0$, el efecto de la moneda mala se compensa con el de la buena y la probabilidad de ganar resulta finalmente $1/2$, por lo que el juego B también es justo.

La cadena de Markov asociada al juego B es la dada en la figura 2.5:

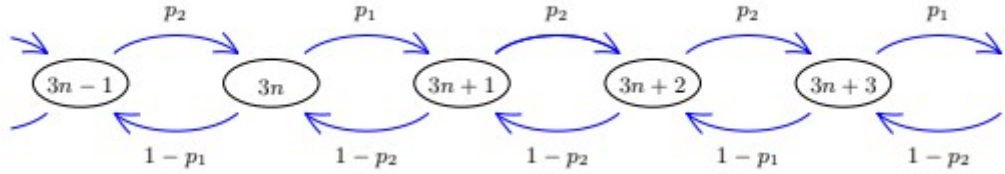


Figura 2.5: Cadena de Markov para el juego B . Figura extraída de [18].

Esta cadena consiste en infinitos subsistemas periódicos, cada uno de longitud M , donde aquí se ha tomado, como en la paradoja de Parrondo original, $M = 3$. En esta cadena podemos observar claramente la dependencia del estado en el que nos encontremos, pues la probabilidad que nos hace pasar de un estado múltiplo de 3 al siguiente es $p_1 = p_m$, mientras que las de los demás estados son $p_2 = p_b$. Análogamente, se retrocede al estado anterior con probabilidades $1 - p_1$ y $1 - p_2$, dependiendo de en qué estado nos encontremos.

La matriz de transición asociada a esta cadena de Markov es la siguiente:

$$\mathcal{P}_B = \begin{pmatrix} 0 & p_1 & 0 & 0 & 0 & 0 & (1-p_1) \\ 1-p_2 & 0 & p_2 & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & 1-p_2 & 0 & p_2 & 0 & 0 \\ 0 & 0 & 0 & 1-p_1 & 0 & p_1 & 0 \\ 0 & 0 & 0 & 0 & 1-p_2 & 0 & \ddots \\ (p_2) & 0 & 0 & 0 & 0 & \ddots & \ddots \end{pmatrix}$$

De esta manera, en cada fila múltiplo de M las probabilidades de ganar y perder son, respectivamente, p_1 y $1 - p_1$. Por analogía con lo realizado en el juego A , los términos entre paréntesis en $\mathcal{P}_B$ se han incluido por completitud.

Para estudiar la cadena de Markov del juego B de una forma más sencilla, podemos centrarnos únicamente en cada uno de estos subsistemas que se repiten de manera periódica. Para ello, definimos el subsistema como

$$Y(t) \equiv X(t) \bmod M$$

Aunque haciendo esto no podemos conocer el verdadero valor del capital, sí que podemos conocer si este tenderá a crecer, mantenerse constante o decrecer, que es lo que verdaderamente nos importa. Así construida, la cadena de Markov asociada a este subsistema tiene M estados, que son $\{0, \dots, M - 1\}$, y es cíclica, de manera que el estado siguiente al estado $M - 1$ es el 0 y viceversa. Entonces, la cadena de Markov asociada a $Y(t)$ es la de la figura 2.6:

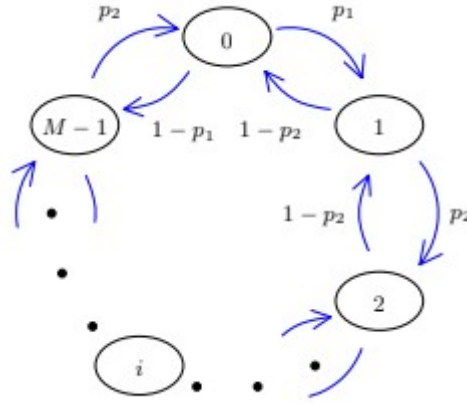


Figura 2.6: Cadena de Markov reducida para el juego B . Figura extraída de [18].

La matriz de transición asociada a esta cadena de Markov es la siguiente:

$$\mathcal{P}_B = \begin{pmatrix} 0 & p_1 & 0 & 0 & 1 - p_1 \\ 1 - p_2 & 0 & p_2 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 1 - p_2 & 0 & p_2 \\ p_2 & 0 & 0 & 1 - p_2 & 0 \end{pmatrix}$$

De esta manera, hemos reducido la matriz a una de tamaño $M \times M$ y ahora los términos que antes estaban entre paréntesis sí que son importantes, ya que se emplean al trabajar módulo M .

Observación 2.2.6. Este mismo enfoque de usar estados periódicos también puede ser empleado para el juego A , de manera que simplemente habría que realizar la sustitución $p_1 = p_2 = p$. Representar el juego A y el juego B mediante matrices finitas en lugar de matrices infinitas nos permitirá realizar cálculos sobre ellas más fácilmente.

Hay que tener cuidado porque a simple vista podríamos malinterpretar el juego B como un juego ganador. Por ejemplo, con $M = 3$, un razonamiento erróneo asumiría que las probabilidades de tomar los valores de $X(t) \bmod 3 = \{0, 1, 2\}$ valen $\frac{1}{3}$ cada una. Es decir, bajo este razonamiento erróneo la moneda mala se usaría $\frac{1}{3}$ de las veces y la moneda buena el otro $\frac{2}{3}$ de las veces. A partir de las parametrizaciones empleadas y tomando $\epsilon = 0$, la probabilidad de ganar sería de $\frac{1}{3} \frac{1}{10} + \frac{2}{3} \frac{3}{4} = \frac{16}{30}$, con lo que estaríamos ante un juego ganador, lo cual es incorrecto pues el juego B es en realidad un juego justo, como veremos a continuación.

En el juego B se usan dos monedas, una mala que se lanza cuando el capital es múltiplo de 3, y otra buena que se utiliza en caso de que esto no sea así. La ganancia esperada del juego B depende, por tanto, de la frecuencia con que se utilice la moneda mala o, dicho de otra manera, de la probabilidad $\pi_0(t)$ de que la variable aleatoria $X(t)$ sea un múltiplo de 3.

A simple vista, se razona que cuando el capital no es múltiplo de 3 es más probable el movimiento hacia la derecha que hacia la izquierda, es decir, pasar al estado siguiente, mientras que si es múltiplo de 3 lo más probable será un movimiento hacia la izquierda. Así, el capital pasa mucho más tiempo saltando hacia delante y hacia atrás entre un estado múltiplo de 3 y su estado inmediatamente anterior en la cadena del que lo haría si el movimiento fuera completamente aleatorio. Esto nos hace sospechar que $\pi_0(t)$ será algo mayor que $1/3$.

Para el caso en el que juguemos aleatoriamente a los juegos A y B , podemos ver que en el juego A el movimiento del capital sí es prácticamente aleatorio, pues ϵ es un sesgo pequeño, de manera que cuando se juega A durante algunos turnos, la probabilidad $\pi_0(t)$ va acercándose cada vez más a $1/3$.

Esto nos hace pensar que jugar A reduce la probabilidad de que al jugar B de nuevo haya que utilizar la moneda mala. Esta observación es fundamental, ya que nos hace entender un poco la razón por la que se da la paradoja, pues a pesar de ser un juego perdedor, el juego A refuerza el efecto de la moneda buena presente en B , lo que proporciona al juego aleatorio un comportamiento ganador.

Veamos cuánto vale dicha probabilidad $\pi_0(t)$ y con ello veremos que efectivamente no es $1/3$, sino un valor ligeramente mayor. Si consideramos únicamente el juego B , la probabilidad de que el jugador gane en el turno t se expresa como:

$$p_{\text{ganar}B}(t) = \pi_0(t)p_m + (1 - \pi_0(t))p_b$$

donde recordemos que $\pi_0(t)$ es la probabilidad de que $X(t)$ sea múltiplo de 3, $p_m = p_1$ es la probabilidad de ganar con la moneda mala y $p_b = p_2$ es la probabilidad de ganar con la moneda buena.

Obtenida la probabilidad de ganar el juego B en el turno t , se puede calcular la ganancia media en el turno t sustituyendo p por $p_{\text{ganar}B}$ en la expresión de la ganancia del juego A , $2p - 1$, luego para el juego B se tendría que la ganancia media es $2p_{\text{ganar}B} - 1$.

Como el juego B viene determinado por una dinámica periódica, en realidad basta con conocer el estado $Y(t) \equiv X(t) \bmod 3$ para describir la evolución del sistema y por lo tanto ver si la ganancia es positiva, negativa o nula. $Y(t)$ así definido es un proceso estocástico que puede tomar tres valores posibles: 0, 1 o 2, dependiendo del valor del capital módulo 3. Así definido, tenemos que $\pi_0(t)$ es la probabilidad de que $Y(t) = 0$. Se define de manera análoga $\pi_1(t)$ y $\pi_2(t)$ como la probabilidad de que $Y(t) = 1$ o $Y(t) = 2$ respectivamente.

$Y(t)$ viene entonces representado por una cadena de Markov, y, por ello, basta conocer el estado $Y(t)$ y la matriz de transición para determinar con qué probabilidad encontraremos cada uno de los tres estados posibles en el turno siguiente. De esta manera podremos escribir una ecuación para la evolución de las probabilidades $\pi_0(t)$, $\pi_1(t)$ y $\pi_2(t)$. A partir de las reglas del juego B podemos calcular la probabilidad p_{ij} de transición entre el estado i y el estado j , que para $M = 3$ son:

$$\mathcal{P}_B = \begin{pmatrix} 0 & p_m & 1 - p_m \\ 1 - p_b & 0 & p_b \\ p_b & 1 - p_b & 0 \end{pmatrix}$$

A partir de esta matriz, al estado j se puede llegar en el turno $t + 1$ desde cualquiera de los otros estados i en t con probabilidad (dada por la probabilidad de transición) p_{ij} , de modo que la probabilidad $\pi_j(t + 1)$ viene dada por la unión de todos esos sucesos independientes, es decir la suma

$$\pi_j(t + 1) = \sum_{i=0}^2 p_{ij} \pi_i(t)$$

Utilizando el vector de probabilidad $\pi(t) = (\pi_0(t), \pi_1(t), \pi_2(t))$, la ecuación anterior se escribe

$$\pi(t + 1) = \pi(t) \mathcal{P}_B$$

Observamos que la cadena de Markov reducida mostrada en la figura 2.6 que define el juego B es finita, pues tiene M estados. Además, todos sus estados son recurrentes positivos, intercomunican y, para el caso específico en que $M = 3$, los estados son aperiódicos, pues una vez hemos salido de un estado podemos volver a ese mismo estado pasando a uno de sus estados adyacentes y volviendo al estado original o completando una vuelta entera en la cadena de Markov, con lo que en vez de volver al estado inicial en 2 pasos se llegaría en $M = 3$ pasos.

Por lo tanto, en este caso podemos aplicar los teoremas 1.0.1 y 1.0.2 de los preliminares sobre la existencia de la distribución estacionaria y su coincidencia con la distribución límite. De la misma forma, los teoremas siguen siendo válidos para la cadena del juego A sin más que sustituir p_m y p_b por p .

Empleando *Mathematica* con el código 7.2.1, hemos calculado la distribución estacionaria del juego B con $\epsilon = 0$, obteniendo que es:

$$\pi_B^{st} = \left(\frac{5}{13}, \frac{2}{13}, \frac{6}{13} \right)$$

Observamos que efectivamente estamos ante una distribución de probabilidad ya que $\pi_0 + \pi_1 + \pi_2 = 1$.

Como habíamos mencionado antes, ahora obtenemos que la probabilidad π_0 es ligeramente mayor que $\frac{1}{3}$, y la probabilidad π_1 es sustancialmente menor que las demás ya que el movimiento más probable en la cadena es la alternancia entre los estados $Y = 2$ e $Y = 0$.

Observación 2.2.7. Para analizar el juego B de manera correcta hemos empleado la teoría de cadenas de Markov, con lo que se observa que la probabilidad de estar en cada estado no es de $\frac{1}{3}$, sino de $\frac{5}{13}$, $\frac{2}{13}$ y $\frac{6}{13}$ respectivamente, y así la probabilidad de ganar calculada ahora ya de manera correcta sería $\frac{5}{13} \cdot \frac{1}{10} + \frac{2}{13} \cdot \frac{3}{4} + \frac{6}{13} \cdot \frac{3}{4} = \frac{1}{2}$, con lo que comprobamos que el juego es efectivamente justo.

Observación 2.2.8. Para cualquier distribución inicial dada (a, b, c) , con $a, b, c \geq 0$ y $a + b + c = 1$, la distribución de probabilidades de cada uno de los estados en el infinito, esto es, la distribución límite, jugando únicamente el juego B es $(\frac{5}{13}, \frac{2}{13}, \frac{6}{13})$.

Hasta ahora hemos estudiado el caso sencillo en el que $\epsilon = 0$. Pero, ¿qué sucede para $\epsilon > 0$? De nuevo, calculando la distribución estacionaria empleando *Mathematica* con el código del anexo 7.2.1, y haciendo un desarrollo de Taylor de orden dos de la misma en $\epsilon = 0$, obtenemos que la primera componente de la solución estacionaria es

$$\pi_{B0}^{st} = \frac{5}{13} - \frac{440}{2197}\epsilon - \frac{34080}{371293}\epsilon^2 + O(\epsilon^3) \simeq 0.384612 - 0.2002\epsilon - 0.0918\epsilon^2$$

Análogamente, la segunda componente de la solución estacionaria es

$$\pi_{B1}^{st} = \frac{2}{13} + \frac{188}{2197}\epsilon + \frac{97648}{371293}\epsilon^2 + O(\epsilon^3) \simeq 0.1538 + 0.0856\epsilon + 0.2630\epsilon^2$$

Y la tercera componente es

$$\pi_{B2}^{st} = \frac{6}{13} + \frac{252}{2197}\epsilon - \frac{63568}{371293}\epsilon^2 + O(\epsilon^3) \simeq 0.4615 + 0.1147\epsilon - 0.1712\epsilon^2$$

La probabilidad de ganar al jugar B durante un número suficiente de turnos es

$$\begin{aligned} p_{\text{ganar}B} &\simeq (0.384612 - 0.2002\epsilon - 0.0918\epsilon^2)(0.1 - \epsilon) + (0.1538 + 0.0856\epsilon + 0.2630\epsilon^2)(0.75 - \epsilon) + \\ &\quad + (0.4615 + 0.1147\epsilon - 0.1712\epsilon^2)(0.75 - \epsilon) \simeq \\ &\quad \simeq \frac{1}{2} - 0.869822\epsilon + 0.0596618\epsilon^2 + O(\epsilon^3) \end{aligned}$$

Observamos que, efectivamente, este valor es menor que $1/2$ si $\epsilon > 0$, lo que prueba que B es un juego justo si $\epsilon = 0$ y perdedor si $\epsilon > 0$.

El juego A puede analizarse de manera similar. La matriz de transición se escribiría de la siguiente manera:

$$\mathcal{P}_A = \begin{pmatrix} 0 & p & 1-p \\ 1-p & 0 & p \\ p & 1-p & 0 \end{pmatrix}$$

y la distribución estacionaria para el juego A está dada por

$$\pi_A^{st} = \pi_A^{st} \mathcal{P}_A$$

Analizando el funcionamiento del juego A , podemos darnos cuenta de que las probabilidades de transición no dependen del capital $Y(t)$ y, en consecuencia, la solución estacionaria debe tener todas sus componentes iguales, esto es, $\pi_0^{\text{st}} = \pi_1^{\text{st}} = \pi_2^{\text{st}}$. Como la suma total de estos tres elementos debe ser igual a 1, se tiene que

$$\pi_A^{\text{st}} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right)$$

Con esto vemos que jugar A acerca $\pi_0(t)$ hacia $\frac{1}{3}$ y, como $\frac{1}{3}$ es menor que $\frac{5}{13}$, efectivamente jugar A reduce la probabilidad de que el jugador se encuentre en la moneda mala cuando vuelva a jugar B .

Para definir la **combinación aleatoria** de ambos juegos, se emplea el parámetro $\gamma \in [0, 1]$, que determina la probabilidad de jugar A , siendo entonces $1 - \gamma$ la probabilidad de jugar B . En nuestro caso, para construir la combinación de los juegos de manera que la aparición de ambos sea equiprobable, tomaremos $\gamma = 1/2$. En este caso, denotaremos a dicha combinación aleatoria como $A + B$, y la denominaremos **mezcla aleatoria** de los juegos A y B . Es importante tener en cuenta esta diferencia entre combinación aleatoria y mezcla aleatoria para futuros resultados.

Observación 2.2.9. Si se desea simular los juegos A, B y su mezcla aleatoria $A + B$, se puede emplear el código desarrollado en 7.1.2.

La combinación aleatoria de ambos juegos se describe como

$$\pi(t+1) = \pi(t)(\gamma \mathcal{P}_A + (1 - \gamma) \mathcal{P}_B)$$

puesto que en cada turno la probabilidad de ganar en el estado $Y(t)$ se calcula como la probabilidad de jugar A (que es γ) por la probabilidad de ganar en A , que es p_A más la probabilidad de jugar B (que es $1 - \gamma$) por la probabilidad de ganar jugando B (que es p_m o p_b según corresponda al estado). Esto nos permite ver que la cadena de Markov asociada a la combinación aleatoria de A y B es la análoga a la de los juegos originales pero sustituyendo las probabilidades en ellos por estas probabilidades, luego los teoremas 1.0.1 y 1.0.2 siguen siendo válidos para la combinación aleatoria con cualquier valor de γ en el intervalo $[0, 1]$.

Como ya hemos mencionado, consideraremos ambos juegos equiprobables y así en la mezcla aleatoria se tiene:

$$\pi(t+1) = \pi(t) \frac{1}{2} [\mathcal{P}_A + \mathcal{P}_B]$$

A partir de esta ecuación sobre las distribuciones de probabilidad, se tiene que la distribución estacionaria verifica

$$\pi_{A+B}^{\text{st}} = \pi_{A+B}^{\text{st}} \frac{1}{2} [\mathcal{P}_A + \mathcal{P}_B]$$

De nuevo, haciendo los cálculos con *Mathematica* con el código en 7.2.1 se observa que la primera componente de la solución estacionaria en el juego $A + B$ es

$$\pi_{A+B0}^{\text{st}} \simeq 0.345557 - 0.0972386\epsilon - 0.02094\epsilon^2$$

A su vez, la segunda componente de la solución estacionaria es

$$\pi_{A+B1}^{st} \simeq 0.253879 + 0.0453091\epsilon + 0.109628\epsilon^2$$

Y la tercera componente de la solución estacionaria es

$$\pi_{A+B2}^{st} \simeq 0.400564 + 0.0519296\epsilon - 0.0886881\epsilon^2$$

La probabilidad de ganar en la mezcla aleatoria es, en el régimen estacionario,

$$\begin{aligned} p_{A+B} &= \frac{1}{2}p_A + \frac{1}{2}[\pi_{0A+B}^{st}p_m + (1 - \pi_{0A+B}^{st})p_b] \simeq \\ &\simeq \frac{1}{2}\left(\frac{1}{2} - \epsilon\right) + \frac{1}{2}[(0.345557 - 0.0972386\epsilon - 0.02094\epsilon^2)\left(\frac{1}{10} - \epsilon\right) + \\ &\quad + (1 - (0.345557 - 0.0972386\epsilon - 0.02094\epsilon^2))\left(\frac{3}{4} - \epsilon\right)] \end{aligned}$$

Desarrollando hasta segundo orden, la probabilidad de ganar en la mezcla aleatoria es

$$p_{A+B} \simeq 0.512694 - 0.968397\epsilon + 0.00680551\epsilon^2$$

que es mayor que $1/2$ para ϵ suficientemente pequeño.

Esta es la esencia de la paradoja: el juego A , a pesar de que en él se esté jugando con una moneda con una probabilidad de perder mayor que la probabilidad de ganar, altera las frecuencias con que se juegan la moneda mala y la buena en el juego B , con lo que se consigue emplear la moneda buena lo suficiente como para invertir la naturaleza perdedora existente en ambos juegos.

Observación 2.2.10. Un acercamiento a estos resultados de carácter más analítico puede encontrarse en [19] y [20], donde se calculan las condiciones para que se dé la paradoja de Parrondo en función de las probabilidades p, p_m, p_b . Para ello, se calcula de manera análoga a como se ha hecho en este trabajo la probabilidad de ganar del juego A , del juego B y de la mezcla aleatoria en función de p, p_b, p_m , y se impone que las dos primeras sean menores que $\frac{1}{2}$ y que la tercera sea mayor que $\frac{1}{2}$, de manera que se obtienen una serie de desigualdades sobre p, p_b, p_m necesarias para que la paradoja se verifique. Este enfoque lo llevaremos a cabo posteriormente para el caso de los juegos cooperativos, como se verá más adelante.

Una pregunta interesante que plantear es hasta qué valor de ϵ la mezcla de los juegos es ganadora. Ya hemos probado que para todo $\epsilon > 0$ el juego A y el juego B son perdedores, y hemos probado también que para un ϵ pequeño y positivo la mezcla de esos juegos es ganadora. Pero, ¿cómo de pequeño debe ser ese ϵ ? Para ello, hay que estudiar cuándo

$$0.512694 - 0.968397\epsilon + 0.00680551\epsilon^2 \geq 0.5$$

Resolviendo con *Mathematica*, observamos que el valor máximo de ϵ es aproximadamente 0.0131093, luego la paradoja de Parrondo se verifica siempre que $\epsilon \in [0, 0.0131093]$.

De esta manera, si $\epsilon > 0$ la probabilidad ambos juegos resultan perdedores. Sin embargo, si en cada turno se elige un juego al azar o incluso si se juegan los juegos A y B

de forma periódica, siguiendo una secuencia predeterminada como $ABBABB\dots$, entonces el capital medio $\mathbb{E}(X(t))$ crece con el número de turnos, incluso para $\epsilon > 0$ siempre que este sea lo suficientemente pequeño. Para observar esto, se han realizado algunas simulaciones en *Python* con los códigos que aparecen en 7.1.5, obteniendo los resultados que se muestran en las figuras 2.7, 2.8 y 2.9:

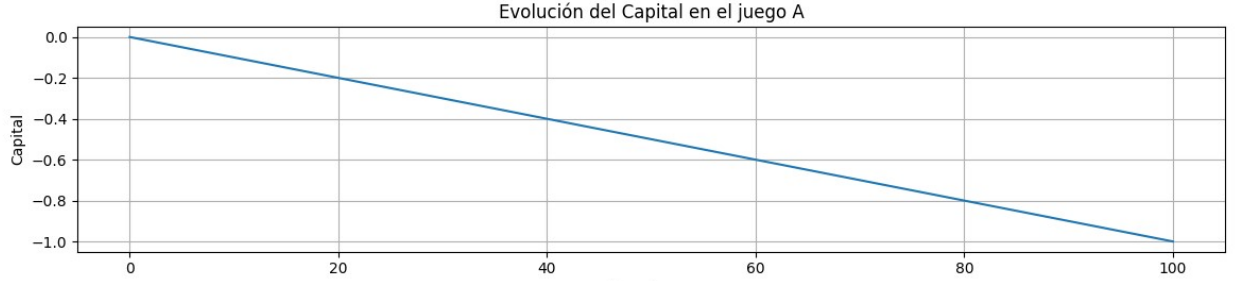


Figura 2.7: Evolución del capital promedio para el juego A . Elaboración propia.

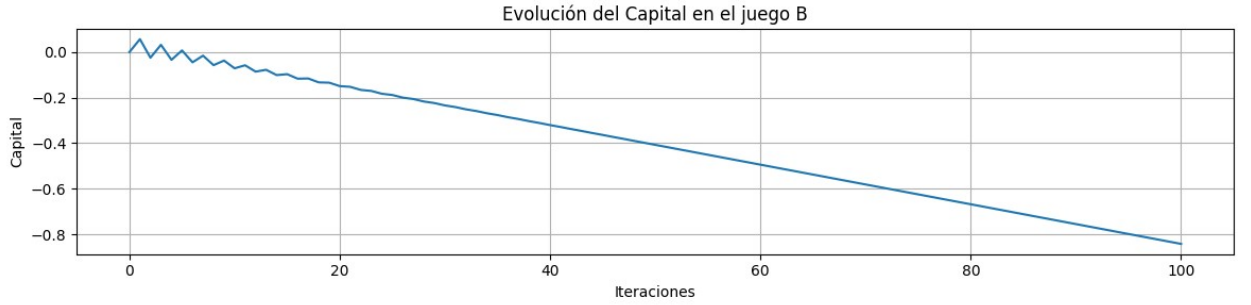


Figura 2.8: Evolución del capital promedio para el juego B . Elaboración propia.

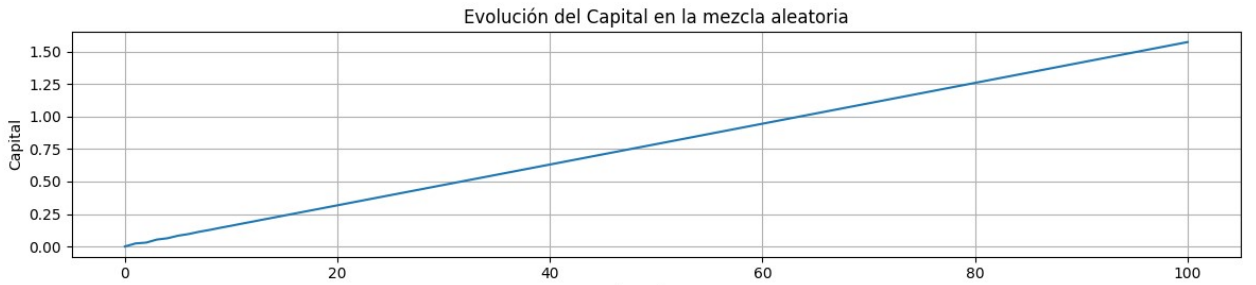


Figura 2.9: Evolución del capital promedio para la mezcla aleatoria. Elaboración propia.

Para realizar estos gráficos se ha considerado un valor de $\epsilon = 0.005$ y 100 iteraciones. La distribución inicial elegida es $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. La razón de esta elección ha sido que, al comenzar a jugar, no hay motivo para suponer que el capital del jugador tiene más probabilidad de ser un múltiplo de 3, un múltiplo de 3 más 1 o un múltiplo de 3 más 2, lo que hace que las consideremos todas iguales entre sí.

Esto puede observarse en la gráfica del juego A , donde como su distribución estacionaria es $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, no se generan esas distorsiones iniciales que pueden ser observadas en el juego $A + B$ y, sobre todo, en el juego B .

Observación 2.2.11. La paradoja de Parrondo se da en este caso porque ambos juegos A y B no son mutuamente excluyentes, sino que están ligados mediante el capital. El juego A , aun teniendo una probabilidad de ganar menor que la de perder, redistribuye las frecuencias con las que se juegan la moneda buena y mala en B , de manera que hace que la moneda buena se juegue más en la mezcla aleatoria y esta sea ganadora.

Observación 2.2.12. Es interesante mencionar que si en vez de elegir los juegos A o B de manera aleatoria permitiésemos al jugador decidir qué juego quiere jugar en cada turno, el jugador escogería el juego A cuando el capital es múltiplo de 3 y el B en otro caso.

Siguiendo el razonamiento hecho en [18], podemos observar también que, dependiendo de la cantidad inicial con la que contemos al empezar a jugar a la mezcla aleatoria, al principio del juego es muy probable que ganemos o perdamos una pequeña cantidad de dinero, dependiendo de si nuestra cantidad inicial es o no múltiplo de M . Para ejemplificar esto, se ha hecho el gráfico en *Python* que se observa en la figura 2.10:

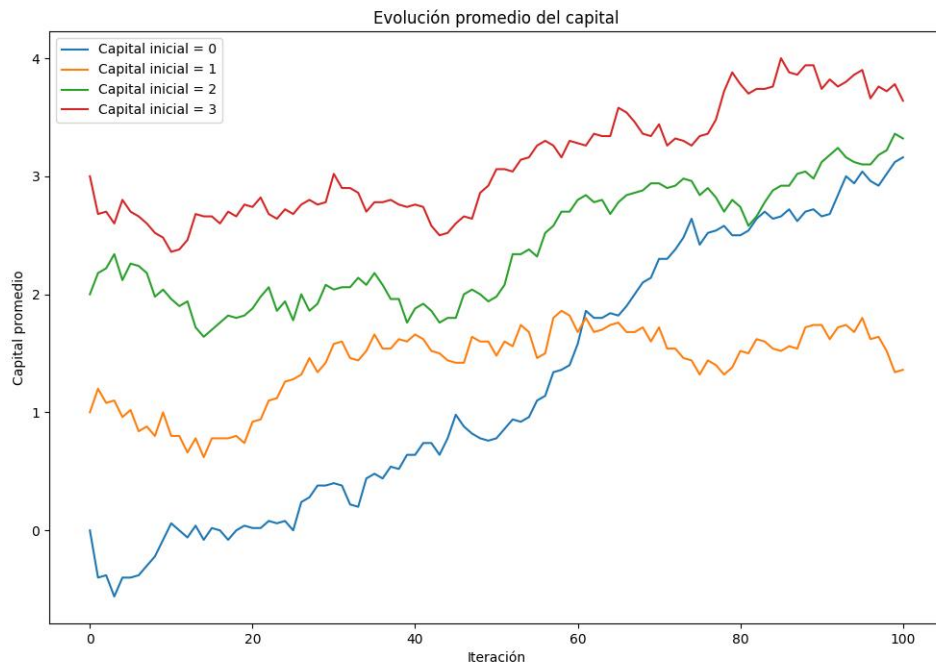


Figura 2.10: Comparación del capital promedio en la mezcla aleatoria en función del capital inicial. Elaboración propia.

Para hacer el gráfico, se ha empleado el código que aparece en 7.1.8. Como se puede deducir de este gráfico, en este caso $M = 3$ ya que si empezamos con un capital inicial 0 o 3, en primera instancia el capital tiende a disminuir, lo cual no sucede si empezamos con un capital inicial de 1 o 2. Para realizar el gráfico se ha hecho la simulación con $\epsilon = 0.005$ y se han promediado los resultados sobre 100 simulaciones.

Observación 2.2.13. Al igual que existe la paradoja de Parrondo, también podríamos hablar de una “antiparadoja de Parrondo”, en el sentido de que combinar dos juegos ganadores puede dar lugar a un juego perdedor. La explicación detrás de esto es exactamente la misma que la de la paradoja original, y para entenderlo podemos tratar de ver la situación como si nosotros fuéramos las personas que ofrecemos jugar al jugador. Desde nuestro punto de vista, le ofrecemos dos juegos ganadores (que serán juegos perdedores para él), y el jugador alternando entre dichos juegos es capaz de crear un juego ganador (y por lo tanto perdedor desde nuestro punto de vista).

En la figura 2.11 se representa el capital medio en función de la secuencia determinística (a, b) con la que se juega a los juegos A y B , donde se juega A un número a de veces y después B un número b de veces:

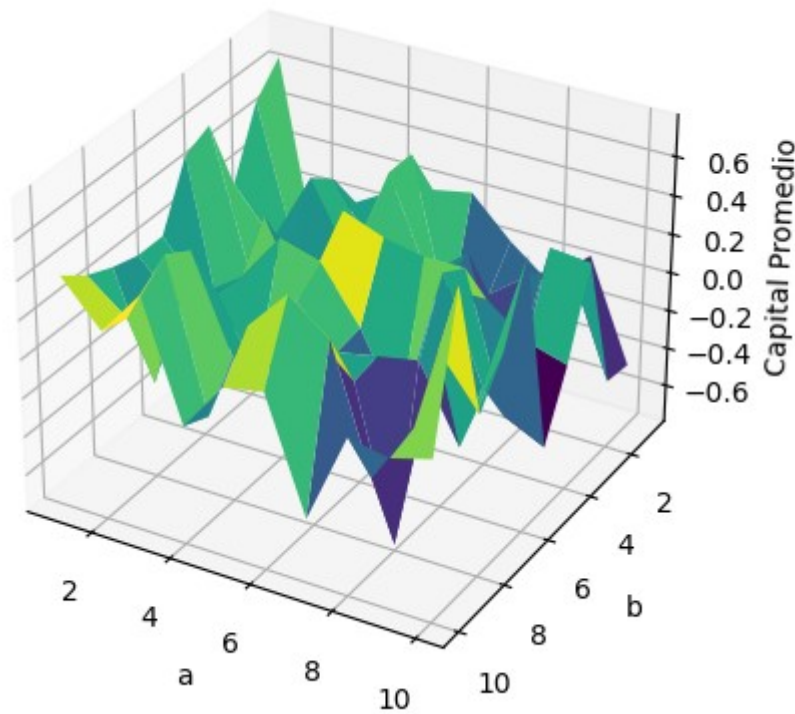


Figura 2.11: Comparación del capital promedio esperado en función de a y b . Elaboración propia.

Para hacer este gráfico, se ha empleado $M = 3$, $\epsilon = 0.005$ y los parámetros originales. El resultado se ha promediado sobre 100 simulaciones, y el código empleado está en 7.1.9.

En el gráfico observamos que, aunque de manera estocástica, cuanto mayor sea el número de turnos para alternar entre A y B , el capital promedio esperado es más bajo. Esto es debido a que, como ya hemos comentado, lo que provoca la ganancia en la mezcla aleatoria es la alternancia entre los juegos, mientras que jugar muchas veces seguidas al mismo juego suele ser perjudicial para el capital. Observamos que la mayor tasa de ganancias es la del juego $(1, 2)$.

Estudiemos este juego. Es decir, supongamos que jugamos los juegos A y B en el orden $ABBABB\dots$. Supongamos también que, habiendo jugado B dos veces seguidas el

jugador tiene un capital múltiplo de 3, es decir, tiene que jugar con una moneda mala. Por ejemplo, imaginemos que su capital es 0. Entonces se juega A , que puede mover al jugador una unidad hacia delante, capital 1, o atrás, capital -1. Si se mueve hacia delante, el jugador tendrá que jugar B con una moneda buena, con lo que es probable que gane una unidad de capital y vuelva a tener que jugar B con una moneda buena. Es decir, hay una alta probabilidad de que en dos turnos el jugador llegue al capital 3, habiendo recorrido así un periodo entero del juego B . En caso contrario, si el jugador con capital 0 juega A y pierde, alcanzará el estado -1. En este caso, la situación más probable para los dos turnos siguientes, en los que tendrá que jugar B , es ganar en la moneda buena y perder en la mala, volviendo a la posición -1. Este análisis ya evidencia cierta rectificación en las dinámicas, puesto que el jugador se mueve mayor distancia hacia estados de capital mayores que menores.

De este modo, los saltos hacia estados menores provocados por A se ven parcialmente anulados por B mientras que los movimientos hacia estados de capital superiores se ven reforzados en los turnos siguientes. Análisis similares muestran que otras secuencias periódicas también producen este efecto de rectificación.

Aplicando el enfoque de [12], podemos obtener la probabilidad de ganar y la ganancia promedio en estado estacionario para secuencias periódicas de los juegos.

Volvamos a considerar la secuencia periódica dada por ABB . ¿Supongamos que en los turnos múltiplos de tres (el periodo de ABB), es decir en los instantes $t = 3n$, con $n = 0, 1, 2, \dots$, se juega A . La ecuación empleada en dicho turno será:

$$\pi(3n + 1) = \pi(3n)\mathcal{P}_A.$$

Sin embargo, en los dos siguientes turnos se tiene que jugar B , por lo que se tiene:

$$\pi(3n + 2) = \pi(3n + 1)\mathcal{P}_B = \pi(3n)\mathcal{P}_A\mathcal{P}_B$$

y

$$\pi(3n + 3) = \pi(3(n + 1)) = \pi(3n + 2)\mathcal{P}_B = \pi(3n)\mathcal{P}_A\mathcal{P}_B\mathcal{P}_B$$

Si nos concentramos en $\mathcal{P}_{ABB} = \mathcal{P}_A\mathcal{P}_B\mathcal{P}_B$, $\pi(3n)$ alcanzará un estado estacionario que denominaremos π^{st0} .

La distribución estacionaria de $\pi(3n + 1)$ y $\pi(3n + 2)$ se obtiene aplicando $\mathcal{P}_A$ y $\mathcal{P}_A\mathcal{P}_B$ a π^{st0} :

$$\pi^{st1} = \pi^{st0}\mathcal{P}_A$$

y

$$\pi^{st2} = \pi^{st0}\mathcal{P}_A\mathcal{P}_B$$

De esta manera, al aplicar $\mathcal{P}_B$ sobre π^{st2} obtenemos π^{st0} de nuevo. Observamos que la distribución no alcanza nunca un estado estacionario estrictamente hablando, sino que realiza un ciclo entre los tres vectores.

Empleando el código 7.1.10 de *Python*, hemos calculado que

$$\pi^{st0} = (0.4429, 0.1743, 0.3828)$$

$$\pi^{st1} = (0.2775, 0.4125, 0.3100)$$

$$\pi^{st2} = (0.3361, 0.1054, 0.5585)$$

A raíz de esto, razonamos que la ganancia media es diferente en los turnos de la forma $3n$, $3n + 1$ y $3n + 2$. Denominémoslas g_0 , g_1 y g_2 respectivamente. Su cálculo se realiza de forma similar a como lo hacíamos para un juego solo, como veremos ahora.

La ganancia g_0 es la ganancia media producida por el juego A, ya que en el turno $3n$ siempre se juega A, por lo que vale:

$$g_0 = 2(p_{\text{ganarA}}) - 1 = 2p - 1$$

La ganancia g_1 es la ganancia correspondiente a jugar B cuando la distribución de probabilidad vale π^{st1} . Podemos entonces calcularla en función del valor de su primera componente π_0^{st1} como:

$$g_1 = 2p_{\text{ganarB}} - 1 = 2(\pi_0^{st1}p_m + (1 - \pi_0^{st1})p_b) - 1$$

y análogamente para g_2 en función de π_0^{st2} tenemos:

$$g_2 = 2(\pi_0^{st2}p_m + (1 - \pi_0^{st2})p_b) - 1$$

Una vez llegamos a un estado estacionario, la ganancia media a lo largo de tres turnos es siempre la misma y coincide con la pendiente media del crecimiento del capital:

$$g_{ABB} = \frac{1}{3}(g_0 + g_1 + g_2)$$

Realizando la simulación en *Python*, se ha obtenido el gráfico de la figura 2.12:

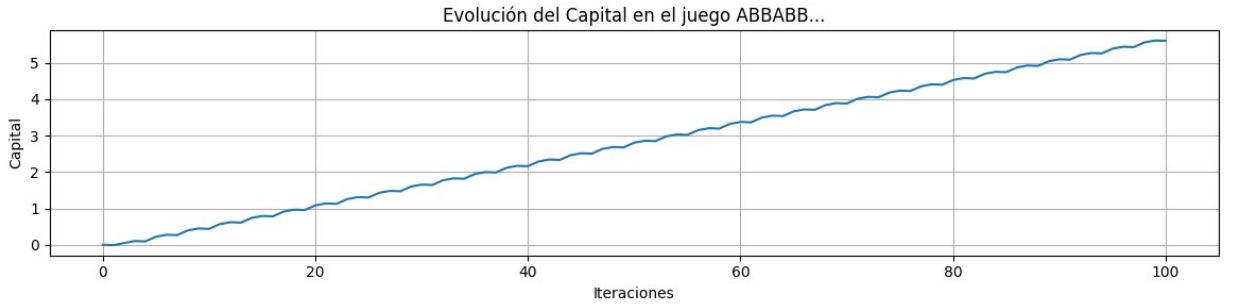


Figura 2.12: Evolución del capital según la secuencia ABBABB... Elaboración propia.

Observamos que la distribución inicial tarda muy pocas iteraciones en converger y se genera un juego ganador. Además, observamos que el capital crece de manera periódica, como es lógico.

Para cerrar este capítulo, podemos interpretar ahora la paradoja de Parrondo en los términos con los que empezamos el mismo, relacionándola con el efecto *ratchet*. Para hacer esto, podemos pensar en el capital $X(t)$ como la posición de una partícula browniana con un movimiento unidimensional. De esta manera, el comportamiento del capital en el

juego B recuerda al movimiento de la partícula browniana bajo la acción del potencial asimétrico.

Para ejemplificar esta relación de manera gráfica tenemos la figura 2.13. En ella mostramos la distribución de la moneda mala y la buena en ese movimiento unidimensional y lo comparamos con el potencial de la *flashing ratchet*:

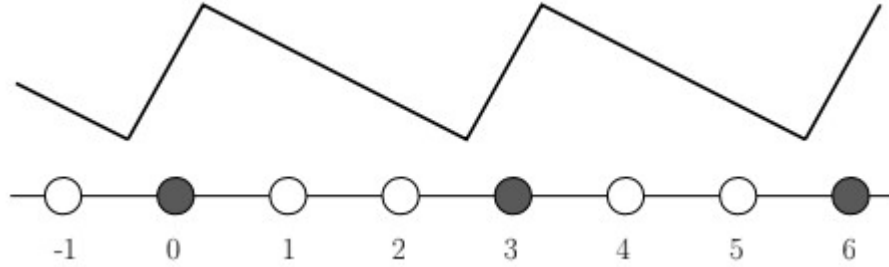


Figura 2.13: Comparación de la distribución de las monedas y el potencial de la *flashing ratchet*. Figura extraída de [12].

Observamos que la moneda mala, representada por los puntos oscuros, es el análogo de la fuerza hacia la izquierda en el tramo corto del potencial mientras que las dos monedas buenas seguidas, representadas por los puntos blancos, se asemejan a la fuerza hacia la derecha actuando en el tramo largo del potencial.

Cuando el jugador se encuentra en una posición oscura, es decir, su capital es múltiplo de 3, tiene que lanzar la moneda mala y la probabilidad de que este gane es muy baja, siendo el movimiento hacia atrás bastante más probable. En los puntos blancos sucede justo la situación opuesta.

El potencial tiene un tramo corto de fuerza hacia la izquierda y un tramo largo de fuerza hacia la derecha. Análogamente, el juego B tiene una moneda mala y dos buenas cada tres estados, que se corresponderían con el tramo corto y el largo respectivamente. De forma similar a como ocurría en la *flashing ratchet*, el juego A y el juego B se combinan para dar lugar a una dinámica ganadora.

Capítulo 3

Juegos dependientes de la historia y juegos cooperativos

Hasta ahora, considerábamos un juego A con probabilidad p de ganar y por tanto probabilidad $1 - p$ de perder. Así, este tipo de juego podía ser entendido mediante el lanzamiento de una moneda sesgada.

Para jugar el juego B , observábamos nuestro capital. Si este era múltiplo de un entero M , la probabilidad de ganar era de p_m , y si no de p_b . Podríamos modelizar este juego mediante el uso de dos monedas, de probabilidades p_m y p_b , viniendo determinada la moneda que lanzamos por nuestro capital. De esta forma, ambos juegos pueden ser modelizados a la vez empleando tres monedas.

En este capítulo ampliamos estos juegos, y con ello la paradoja de Parrondo, a dos casos más. En el primero de ellos, el jugador tendrá memoria de su resultado en los últimos dos turnos que ha jugado, de manera que la probabilidad de ganar en un cierto turno dependerá de los resultados en los dos intentos anteriores.

En el segundo de ellos, consideraremos no sólo un jugador sino un conjunto de ellos. De esta manera, la probabilidad de ganar de un determinado jugador vendrá determinada por el último resultado de sus vecinos.

3.1. Juegos dependientes de la historia

En este caso, consideraremos un juego A y un juego B' . El juego A es el que ya conocemos del capítulo anterior. El juego B' , por su parte, se ha definido de manera que las probabilidades empleadas dependen de los dos estados anteriores del jugador, resultando en cuatro probabilidades posibles, p_1, p_2, p_3, p_4 .

La probabilidad p_1 se emplea cuando en los dos turnos anteriores el jugador perdió una unidad de capital, ya fuera jugando al juego A o al juego B' . La probabilidad p_2 se emplea cuando en el turno anterior ganó y en el anterior a ese perdió, p_3 cuando en el turno anterior perdió y en el anterior a ese ganó y p_4 cuando ganó en los dos turnos anteriores.

En la figura 3.1 podemos observar los diagramas de los juegos A y B' , donde W representa una victoria en el juego y L una derrota:

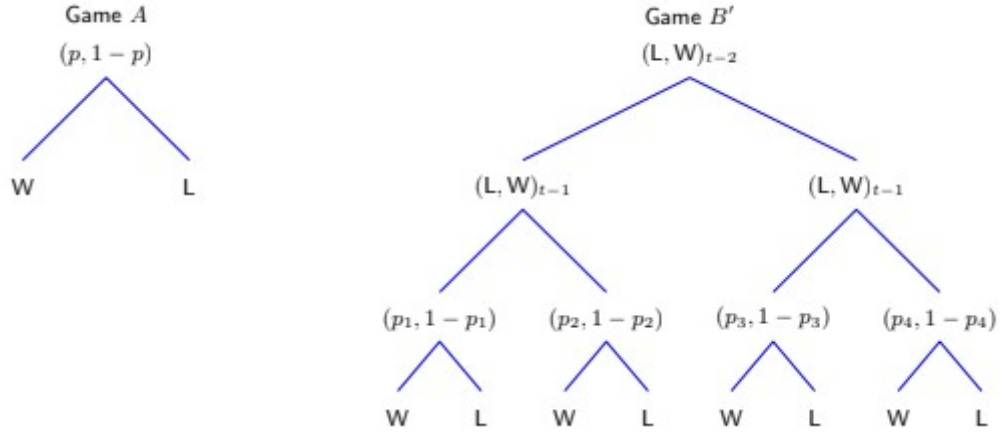


Figura 3.1: Diagrama de árbol de los juegos dependientes de la historia. Figura extraída de [18].

Parametrizaremos las probabilidades de los juegos A y B' de la siguiente manera:

$$p = \frac{1}{2} - \epsilon \quad (3.1)$$

$$p_1 = \frac{9}{10} - \epsilon \quad (3.2)$$

$$p_2 = p_3 = \frac{1}{4} - \epsilon \quad (3.3)$$

$$p_4 = \frac{7}{10} - \epsilon \quad (3.4)$$

Parrondo también estudió este tipo de juegos [19], y esta fue la parametrización original que empleó.

Las probabilidades se han fijado de esta forma porque con esta parametrización los juegos pueden ser considerados justos cuando $\epsilon = 0$, perdedores cuando $\epsilon > 0$ y ganadores cuando $\epsilon < 0$. Para el lector interesado, el código que simula estos juegos se encuentra en 7.1.11.

De nuevo se observan los mismos resultados contraintuitivos que tuvimos en el capítulo anterior. Es decir, cuando se juega a los juegos individualmente el capital decrece, pero la alternancia entre los juegos A y B' da lugar a una esperanza positiva, tanto con la mezcla estocástica como con la determinística. El concepto de juego “ganador” o “perdedor” coincide con el ya explicado en el capítulo anterior.

A partir del anterior diagrama, podemos construir la cadena del juego B' que observamos en la figura 3.2:

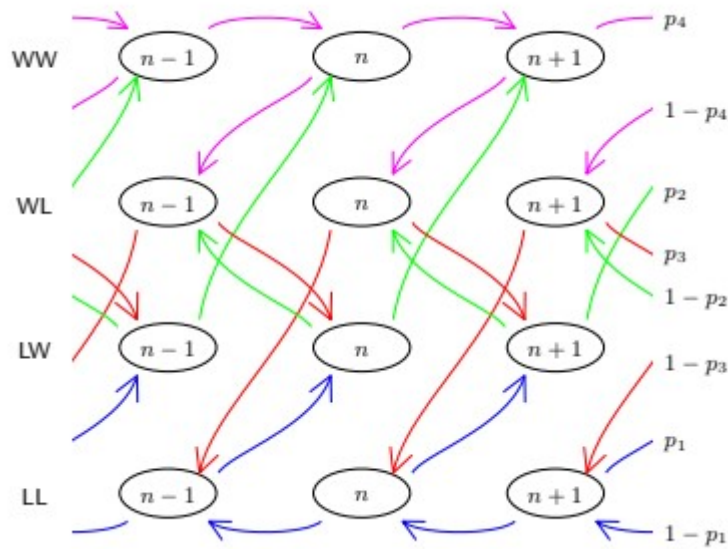


Figura 3.2: Cadena de Markov del juego dependiente de la historia B' . Figura extraída de [18].

Las siglas WW , WL , LW y LL representan las historias de los dos juegos anteriores, y el índice de la columna representa el capital con el que contamos. Por ejemplo, el índice LW representa que hace dos turnos perdimos una unidad de capital y en el turno anterior al presente ganamos una unidad de capital.

Observamos que con esta cadena el próximo estado del capital no depende de los dos resultados anteriores solamente, sino también de en qué columna nos encontremos, es decir, de nuestro capital en dicho momento.

Por esta razón, para simplificar el estudio del juego B' se toman las componentes periódicas de la cadena y se construye así la cadena reducida de la figura 3.3:

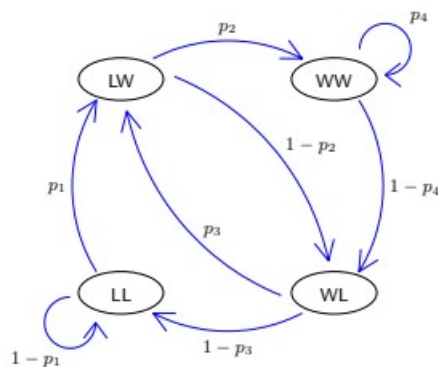


Figura 3.3: Cadena de Markov del juego B' reducida. Figura extraída de [18].

La cadena de Markov correspondiente al juego A se construiría a partir de esta sustituyendo p_1, p_2, p_3, p_4 por p . Como ya sucedía con los juegos originales estudiados en el capítulo anterior, al hacer la reducción no sabemos exactamente cuál es la cantidad de

capital del jugador, pero podemos estudiar lo que verdaderamente nos interesa: si esta decrece, se mantiene o crece.

Observamos que la cadena así definida sí es ergódica y finita luego sabemos por los teoremas 1.0.1 y 1.0.2 que tiene una única distribución estacionaria y que coincide con su distribución límite.

La matriz de transición correspondiente a esta cadena de Markov es:

$$\mathcal{P}_{B'} = \begin{pmatrix} 1-p_1 & p_1 & 0 & 0 \\ 0 & 0 & 1-p_2 & p_2 \\ 1-p_3 & p_3 & 0 & 0 \\ 0 & 0 & 1-p_4 & p_4 \end{pmatrix}$$

La primera fila y la primera columna son las correspondientes a los estados LL , las segundas a LW , las terceras a WL y las cuartas a WW .

Al igual que sucedía anteriormente, al considerar la combinación aleatoria de los juegos las probabilidades del juego resultante vienen dadas por $\gamma p + (1-\gamma)p_i$, con $i = 1, 2, 3, 4$, siendo γ el parámetro de aleatorización que determina la probabilidad de jugar al juego A .

Estamos ahora en disposición de calcular la distribución estacionaria del juego B' , que como ya se ha mencionado coincidirá con la límite. De nuevo empleando *Mathematica* con el código disponible en 7.2.2 y haciendo un desarrollo de Taylor de orden 2 en $\epsilon = 0$ en cada componente hemos obtenido que dicha distribución es

$$\left(\frac{5}{22} + \frac{80\epsilon}{99} + \frac{100\epsilon^2}{891}, \frac{3}{11} + \frac{10\epsilon}{33} - \frac{400\epsilon^2}{297}, \frac{3}{11} + \frac{10\epsilon}{33} - \frac{400\epsilon^2}{297}, \frac{5}{22} - \frac{140\epsilon}{99} + \frac{2300\epsilon^2}{891} \right)$$

De esta manera, para $\epsilon = 0$ se tiene que la distribución estacionaria es

$$\left(\frac{5}{22}, \frac{3}{11}, \frac{3}{11}, \frac{5}{22} \right)$$

Con esta distribución estacionaria, la probabilidad de ganar en el juego B' una vez en estado estacionario es

$$p_{\text{ganar}} = \frac{5}{22} \cdot \frac{9}{10} + 2 \left(\frac{3}{11} \cdot \frac{1}{4} \right) + \frac{5}{22} \cdot \frac{7}{10} = \frac{1}{2}$$

con lo que observamos que efectivamente el juego B' es justo cuando $\epsilon = 0$.

Si hubiéramos asumido erróneamente una probabilidad de $1/4$ en cada estado, la probabilidad de ganar sería

$$p_{\text{ganar}} = \frac{1}{4} \left(\frac{9}{10} + \frac{1}{4} + \frac{1}{4} + \frac{7}{10} \right) = \frac{21}{40} > \frac{1}{2}$$

con lo que el juego B' no hubiera sido justo sino ganador.

Así para $\epsilon = 0$ el juego B' es justo, y perdedor para un ϵ mayor.

Cabe preguntarse de nuevo cuál es la probabilidad de ganar empleando la mezcla aleatoria de los juegos A y B' y, en caso de que esta sea positiva, estudiar el valor máximo

de ϵ para el cual se verifica la paradoja. Para hacer esto, primero es necesario construir la matriz de transición del juego A en este caso, es decir, considerando los cuatro estados mencionados. Para construir dicha matriz, se parte de la matriz del juego B' y se hace en ella la sustitución $p_1 = p_2 = p_3 = p_4 = p = \frac{1}{2} - \epsilon$, de modo que la matriz del juego A en este caso es:

$$\mathcal{P}_A = \begin{pmatrix} 1-p & p & 0 & 0 \\ 0 & 0 & 1-p & p \\ 1-p & p & 0 & 0 \\ 0 & 0 & 1-p & p \end{pmatrix}$$

A partir de $\mathcal{P}_A$ y haciendo un desarrollo de Taylor de orden 2 en $\epsilon = 0$ en cada una de las cuatro componentes podemos calcular la distribución estacionaria del juego A en función de ϵ , obteniendo que esta es:

$$\left(\frac{1}{4} + \epsilon + \epsilon^2, \frac{1}{4} - \epsilon^2, \frac{1}{4} - \epsilon^2, \frac{1}{4} - \epsilon + \epsilon^2 \right)$$

Con esta distribución estacionaria, la probabilidad de ganar el juego A en estado estacionario es

$$1/2 - \epsilon$$

No es difícil comprobar que de nuevo, para $\epsilon = 0$, la distribución estacionaria toma el mismo valor para todos los estados, esto es, viene dada por el vector

$$\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right)$$

y que efectivamente el juego A es justo cuando $\epsilon = 0$ y perdedor cuando $\epsilon > 0$.

Calculando la matriz de la mezcla aleatoria de los juegos de la manera explicada anteriormente, es decir

$$\mathcal{P}_{A+B} = \frac{1}{2}\mathcal{P}_A + \frac{1}{2}\mathcal{P}_B$$

estamos en disposición de calcular la distribución estacionaria de la misma y con ella la probabilidad de ganar de la mezcla.

Realizando los respectivos cálculos con *Mathematica* y disponibles en 7.2.2, hemos obtenido que:

$$\pi^{A+B'} = \left(\frac{100}{429} + \frac{13990\epsilon}{16731} + \frac{328600\epsilon^2}{652509}, \frac{112}{429} + \frac{2440\epsilon}{16731} - \frac{657200\epsilon^2}{652509}, \right. \\ \left. \frac{112}{429} + \frac{2440\epsilon}{16731} - \frac{657200\epsilon^2}{652509}, \frac{35}{143} - \frac{6290\epsilon}{5577} + \frac{328600\epsilon^2}{217503} \right)$$

Calculada la distribución estacionaria, se tiene que la probabilidad de ganar es

$$\frac{217}{429} - \frac{16430\epsilon}{16731} + \frac{328600\epsilon^2}{652509}$$

luego la paradoja de Parrondo se verifica para todo ϵ en el rango $0 < \epsilon < \frac{1}{168}$.

Observación 3.1.1. De nuevo, podemos ver un enfoque más analítico a este tipo de juegos en [19], [33] y [38], donde se calculan las condiciones para que se dé la paradoja de Parrondo en función de las probabilidades p, p_1, p_2, p_3, p_4 siguiendo un enfoque análogo al explicado en la observación 2.2.10.

Observación 3.1.2. El descubrimiento de la existencia de la paradoja de Parrondo en los juegos dependientes de la historia fue una noticia gratamente recibida entre la comunidad científica dedicada al tema, pues permitía la existencia de nuevas aplicaciones de la paradoja ya que en ciertos campos como la biología resultaba más adecuado buscar una aplicación de este tipo de juegos que de los dependientes del capital, pues la conexión de estos juegos con dicho campo científico parecía más natural.

3.2. Juegos cooperativos

Los juegos cooperativos son el tercer caso de juegos en los que se observa la paradoja de Parrondo hasta donde nosotros conocemos. La principal diferencia respecto a los otros dos tipos de juegos es que se juegan por todo un conjunto de jugadores y no por uno solo como en los casos anteriores. De esta manera, el capital total del juego se entenderá como la suma de los capitales individuales de cada uno de los jugadores, y estos evolucionarán dependiendo del estado de sus jugadores vecinos. Como se mencionó en la introducción, este tipo de juegos difieren de la noción clásica de juego empleada en la teoría de juegos ya que aquí no se consideran coaliciones entre los jugadores.

El estado de cada jugador dependerá de si la vez anterior que jugó resultó en una pérdida o en una ganancia de capital, independientemente de haber jugado A o B . Así, en el juego B cada jugador tendrá cuatro posibles probabilidades con las que ganar, dependiendo de los estados de sus vecinos a su izquierda y a su derecha. Además, consideraremos diversas variantes de este tipo de juegos.

3.2.1. Primera variante

Consideremos un conjunto de N jugadores. Cada uno de los jugadores tiene su capital individual en cada turno $X_i(t), i = 1, \dots, N$. Al igual que en los juegos anteriores, el capital de cada uno de estos jugadores evolucionará por la acción de dos juegos posibles, que ahora denotaremos como A y B'' , de manera que el capital total $X(t) = \sum_{i=1}^N X_i(t)$ también se verá afectado.

La dinámica de la mezcla aleatoria de los juegos, denotada $A + B''$, es la siguiente: en cada instante t , un jugador i es elegido aleatoriamente de entre los N . Dicho jugador i juega A o B'' , donde el juego es elegido de manera aleatoria y cada juego tiene una probabilidad de ocurrencia de $\frac{1}{2}$. Si i gana su juego, asignamos a ese jugador la etiqueta “ganador”, de manera que su capital aumenta en una unidad, esto es, $X_i(t+1) = X_i(t) + 1$. En caso contrario, asignamos a ese jugador la etiqueta “perdedor” y $X_i(t+1) = X_i(t) - 1$. La etiqueta asignada no se cambia hasta que el jugador vuelva a ser elegido en un próximo turno. Para jugar solamente A o solamente B'' en vez de jugar la mezcla aleatoria se procederá de manera análoga.

De esta manera, cada jugador puede estar entonces en dos estados posibles: 0 (correspondiente a la etiqueta “perdedor”) y 1 (correspondiente a la etiqueta “ganador”).

Además, cada jugador tendrá a su vez dos vecinos, uno a cada lado. Por ejemplo, los vecinos del jugador 2 son el jugador 1, a su izquierda, y el jugador 3, a su derecha. El proceso está modelizado de tal forma que el vecino a la derecha de N vuelve a ser el jugador 1, es decir, estamos considerando condiciones de frontera periódicas en el conjunto de jugadores. Como norma general, se supone $N \geq 3$ para que así cada jugador tenga un vecino a su izquierda y un vecino a su derecha.

Podemos adaptar las nociones de juego ganador, perdedor y justo a este caso, donde no las aplicaremos sobre cada uno de los capitales individuales sino sobre el capital total de los jugadores. Es decir, estamos ante un juego ganador si el capital total esperado $\mathbb{E}(X(t))$ aumenta con el tiempo, donde $X(t) = \sum_{i=1}^N X_i(t)$. Análogamente para juego perdedor y juego justo de la manera ya conocida.

El juego A es como el que ya conocemos de las secciones anteriores, con una probabilidad p de ser un juego ganador. Por otra parte, el juego B'' puede asignar 4 posibles probabilidades de ganar al jugador i , dependiendo del estado de los dos jugadores vecinos $i+1$ y $i-1$. De esta manera, la probabilidad del jugador i de ganar en el instante t es:

- p_1 , si los jugadores $i-1$ e $i+1$ se etiquetan como “perdedor”.
- p_2 , si $i-1$ se etiqueta como “perdedor” e $i+1$ se etiqueta como “ganador”.
- p_3 , si $i-1$ se etiqueta como “ganador” e $i+1$ se etiqueta como “perdedor”.
- p_4 , si los jugadores en $i-1$ e $i+1$ se etiquetan como “ganador”.

En el diagrama de árbol de la figura 3.4 podemos observar el proceso del juego B'' :

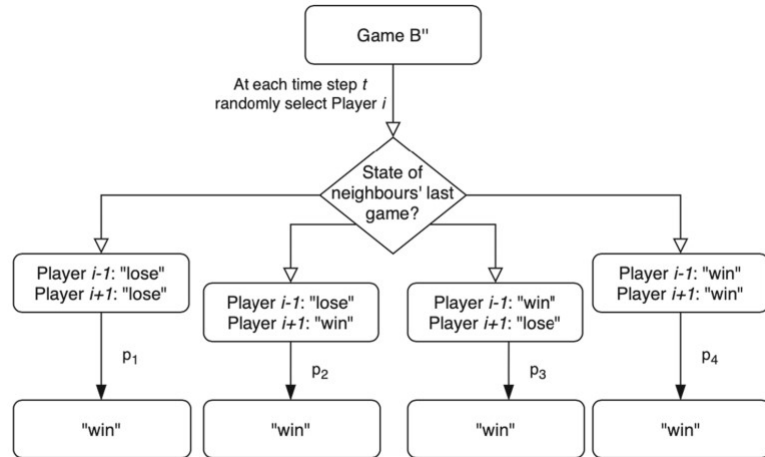


Figura 3.4: Diagrama de árbol del juego cooperativo. Figura extraída de [27].

Como ya mencionamos previamente, el estado de cada jugador puede ser entendido asignando un valor binario a cada uno de ellos. A partir de aquí, el estado de los N jugadores puede ser representado como una cadena binaria $s = (s_1, s_2, \dots, s_N)$, con $s_i \in \{0, 1\}$.

Dicha cadena binaria s puede ser expresada en notación decimal de la forma usual, esto es, a cada $s = (s_1, s_2, \dots, s_N)$ le asignamos el valor decimal $s_N \cdot 2^0 + s_{N-1} \cdot 2^1 + \dots + s_1 \cdot 2^{N-1}$.

Tanto a la cadena binaria como al valor decimal los denotamos por s , y el propio contexto en que se use permitirá diferenciar exactamente a qué nos referimos.

Análogamente, a cada cadena s también le podemos asignar un vector. Veamos detalladamente cómo hacer esto. Si consideramos un conjunto de N jugadores, el valor decimal máximo que se puede asignar a una cadena s es $2^0 + 2^1 + \dots + 2^{N-1}$, correspondiente a la situación en la que todos los jugadores tienen estado 1, es decir, a la cadena $s = (1, 1, \dots, 1)$. De esta manera, el vector asociado a cada valor i entre 0 y $2^0 + 2^1 + \dots + 2^{N-1}$ será un vector de 2^N componentes, donde todas son nulas salvo la posición i , de valor 1. Para cada estado s , denotaremos a su vector asociado como $v(s)$.

Veámoslo ahora con un ejemplo. Para $N = 3$, el estado (011) representa que el jugador 1 actualmente es perdedor, mientras que el jugador 2 y el jugador 3 son ganadores. Su representación decimal sería $s = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 = 3$, y por lo tanto su vector asociado es $v(3) = (0, 0, 0, 1, 0, 0, 0, 0)$.

Sabemos que las probabilidades de ganar en el juego B'' dependen, para cada jugador k , de los estados de los vecinos a su izquierda y a su derecha. Denotaremos a estos dos estados por una pareja $\eta_k = (s_{k-1}, s_{k+1})$. Cuando tengamos la pareja de estados vecinos (0, 0), la probabilidad de ganar el juego B'' del jugador k será de p_0 . Para la pareja (0, 1), tendremos p_1 , para la pareja (1, 0), tendremos p_2 y, finalmente, para la pareja (1, 1), tendremos p_3 .

La función capital $X(t)$ depende de todo el conjunto de jugadores y en cada paso de tiempo incrementa o disminuye su valor en 1 unidad, dependiendo de si el jugador elegido aleatoriamente en esa ronda ha ganado o ha perdido su juego. En cada ronda, cuando el jugador elegido aleatoriamente juega su juego, el estado del conjunto cambia del estado $s(t)$ que tenía en el tiempo t al estado $s(t+1)$ que tendrá en el tiempo $t+1$. De esta manera, el paso del estado $s(t)$ al estado $s(t+1)$ solo supondrá el reemplazo de un 0 a un 1 o de un 1 a un 0 en la cadena $s(t)$.

Si denotamos por $\pi_s(t)$ a la probabilidad de que el conjunto esté en el estado $s(t)$ (o análogamente en el estado $v(s(t))$) entonces la distribución de probabilidad en el tiempo t es $\pi(t) = \sum_{s=0}^{2^N-1} \pi_s(t) v(s)$.

Una vez explicados el juego A , el juego B'' y su aleatorización, hagamos ahora un análisis empleando cadenas de Markov. Primero construiremos la matriz de transición asociada al juego B'' .

Como en cada turno tan solo un jugador i es el que juega y con ello actualiza su estado, este cambio en los estados $s_i(t)$ y $s_{i+1}(t)$ puede ser representado por una distancia de Hamming entre el estado inicial del jugador y el estado final, definida por

$$d_H = \sum_{k=1}^N |s_k(t) - s_k(t+1)|$$

Siguiendo a [30] y ante esta definición de distancia, claramente tenemos dos casos posibles en los valores de la misma.

El primer caso es que $d_H = 0$. Si es así, tenemos que el conjunto de jugadores está en un estado $s(t)$, y el jugador k -ésimo, de estado s_k , ha sido elegido para jugar. El estado $s(t)$ del conjunto también determina el estado de los vecinos del jugador k , $\eta_k = (s_{k-1}, s_{k+1})$. Esta pareja de valores, como se especificó previamente, es la que determinará

las probabilidades que tiene el jugador k -ésimo de ganar en el juego B'' . Como el conjunto de jugadores, que inicialmente está en un estado $s(t)$, puede cambiar al estado $s(t+1) = s(t)$ de N maneras diferentes (como resultado de que uno de los N jugadores pase del estado s_i al mismo estado s_i), la probabilidad de transición es en este caso igual a la suma de probabilidades de los eventos independientes. Cada jugador tiene una probabilidad $\frac{1}{N}$ de ser elegido aleatoriamente, y, una vez elegido, tiene una determinada probabilidad de ganar el juego B'' , que viene determinada por los estados de sus vecinos. Para cada jugador k , denotemos a esta probabilidad como p_{η_k} . De esta manera, si $\eta_k = (0, 0)$ se tiene que $p_{\eta_k} = p_0$, si $\eta_k = (0, 1)$ se tiene que $p_{\eta_k} = p_1$, si $\eta_k = (1, 0)$ se tiene que $p_{\eta_k} = p_2$ y si $\eta_k = (1, 1)$, se tiene que $p_{\eta_k} = p_3$.

Así pues la probabilidad de pasar de un estado $s(t)$ al estado $s(t+1) = s(t)$ es

$$\frac{1}{N} \sum_{k=1}^N w(s_k(t), s_k(t+1))$$

donde el valor de w depende de si el estado $s_k(t)$ es 1 o 0 y del valor de p_{η_k} , esto es:

$$w(s_k(t), s_k(t+1)) = \begin{cases} 1 - p_{\eta_k}, & \text{si } s_k(t) = 0 \\ p_{\eta_k}, & \text{si } s_k(t) = 1 \end{cases}$$

Por ejemplo, para pasar del estado $s(t) = (0101) = 5$ al estado $s(t+1) = (0101) = s(t) = 5$, el primer jugador debe perder en una vecindad dada por $(1, 1)$, el segundo jugador debe ganar en una vecindad $(0, 0)$, el tercer jugador debe perder en una vecindad $(1, 1)$, y el cuarto jugador debe ganar en una vecindad $(0, 0)$. De esta manera, $p_{55} = \frac{1}{4} [(1 - p_{(1,1)}) + p_{(0,0)} + (1 - p_{(1,1)}) + p_{(0,0)}] = \frac{1}{4} [(1 - p_3) + p_0 + (1 - p_3) + p_0]$.

Analicemos ahora el segundo caso, en el que $d_H = 1$. En este caso, el jugador k -ésimo, que ha sido elegido aleatoriamente entre los N jugadores, cambia del estado $s_k(t)$ al estado $s_k(t+1)$, con $s_k(t) \neq s_k(t+1)$, con probabilidad

$$\frac{1}{N} w(s_k(t), s_k(t+1))$$

donde ahora

$$w(s_k(t), s_k(t+1)) = \begin{cases} p_{\eta_k}, & \text{si } s_k(t) = 0 \\ 1 - p_{\eta_k}, & \text{si } s_k(t) = 1 \end{cases}$$

Veámoslo mejor con un ejemplo. Para pasar de $s(t) = 7 = (0111)$ a $s(t+1) = 6 = (0110)$, el cuarto jugador debe ser elegido de manera aleatoria y debe perder en una vecindad $(1, 0)$. Entonces, $p_{76} = \frac{1}{4} [1 - p_{(1,0)}] = \frac{1}{4} (1 - p_2)$. Siguiendo este método, podemos calcular fácilmente la matriz de transición $\mathcal{P}^{B''}$.

Estudiamos ahora la matriz de transición del juego B'' en el caso específico en el que $N = 3$. Razonando como se explicó previamente pero ahora para tres jugadores, se tendría, por ejemplo:

$$\mathcal{P}_{(011),(011)}^{B''} = \mathcal{P}_{3,3}^{B''} = \frac{1}{3} [(1 - p_3) + p_1 + p_2]$$

$$\mathcal{P}_{(011),(111)}^{B''} = \mathcal{P}_{3,7}^{B''} = \frac{1}{3}(p_3)$$

Haciendo estos cálculos, se llega a que la matriz de transición del juego B'' con $N = 3$ es:

$$\mathcal{P}^{B''} = \frac{1}{3} \begin{pmatrix} 3p_0^1 & p_0 & p_0 & 0 & p_0 & 0 & 0 & 0 \\ p_0^1 & Q_2 & 0 & p_1 & 0 & p_2 & 0 & 0 \\ p_0^1 & 0 & Q_2 & p_2 & 0 & 0 & p_1 & 0 \\ 0 & p_1^1 & p_2^1 & Q_1 & 0 & 0 & 0 & p_3 \\ p_0^1 & 0 & 0 & 0 & Q_2 & p_1 & p_2 & 0 \\ 0 & p_2^1 & 0 & 0 & p_1^1 & Q_1 & 0 & p_3 \\ 0 & 0 & p_1^1 & 0 & p_2^1 & 0 & Q_1 & p_3 \\ 0 & 0 & 0 & p_3^1 & 0 & p_3^1 & p_3^1 & 3p_3 \end{pmatrix}$$

En esta matriz, se tiene que $p_i^1 = 1 - p_i$, para $i = 0, 1, 2, 3$, y que $Q_1 = 1 - p_3 + p_1 + p_2$, $Q_2 = 2 - p_2 - p_1 + p_0$. Por las propiedades de estas matrices, se comprueba fácilmente que los elementos de cada fila suman 1, pues cada fila representa una distribución de probabilidad.

A partir de esta matriz, podemos calcular la matriz de transición del juego A , ya que en este las probabilidades de ganar no dependen de los estados de las vecindades, sino que estas son constantes y de valor p . Por lo tanto, para obtener esta probabilidad tan solo hay que reemplazar los valores p_0, p_1, p_2 y p_3 por el valor p , es decir, obtendríamos la matriz

$$\mathcal{P}^A = \frac{1}{3} \begin{pmatrix} 3p^1 & p & p & 0 & p & 0 & 0 & 0 \\ p^1 & 2-p & 0 & p & 0 & p & 0 & 0 \\ p^1 & 0 & 2-p & p & 0 & 0 & p & 0 \\ 0 & p^1 & p^1 & 1+p & 0 & 0 & 0 & p \\ p^1 & 0 & 0 & 0 & 2-p & p & p & 0 \\ 0 & p^1 & 0 & 0 & p^1 & 1+p & 0 & p \\ 0 & 0 & p^1 & 0 & p^1 & 0 & 1+p & p \\ 0 & 0 & 0 & p^1 & 0 & p^1 & p^1 & 3p \end{pmatrix}$$

La matriz de transición del juego aleatorizado es entonces

$$\mathcal{P}^{A+B''} = \frac{1}{2}(\mathcal{P}^A + \mathcal{P}^{B''})$$

Podemos además considerar un vector $v(X)$ cuyas componentes representan el capital medio generado por cada estado del conjunto de jugadores. Como cada elemento del vector está asociado a un determinado estado del conjunto, el vector estará formado por 2^N componentes. A cada una de estas componentes la denotaremos como X_s , donde $s = 0, \dots, 2^N - 1$.

Veamos cómo se genera dicho vector. Dado un determinado estado del conjunto de jugadores $s = (s_1, s_2, \dots, s_N)$, ya sabemos que cada elemento s_i , con $i = 1, \dots, N$, puede tomar el valor 0 o el valor 1. Si un jugador está en el estado 1 sabemos que en su último juego ha conllevado a una ganancia de $+1$ para el capital del conjunto, mientras que si el estado es 0, su último juego habrá conllevado una pérdida de -1 para el capital

del conjunto. X_s es el sumatorio de todas estas ganancias o pérdidas para el capital del conjunto sobre cada uno de los jugadores, dividido por el número de jugadores.

Una vez explicado cómo se generan cada una de las componentes del vector $v(X)$, veamos un poco más en profundidad qué representa cada una. Cada componente puede ser entendida como una medida de cómo de “bueno” es para el capital total de los jugadores estar en dicho estado s . Por ejemplo, la primera componente del vector $v(X)$, denotada como X_0 y correspondiente al estado 0 del conjunto, es un mal estado para el capital, esto es, hace que el capital disminuya, pues para llegar al estado 0 algún jugador debe necesariamente haber perdido su juego. Lo contrario sucede con el estado $2^N - 1$, que es objetivamente bueno para el capital pues necesariamente el último juego jugado debió resultar en una victoria. Para los demás estados, encontramos casos intermedios.

Veamos ahora el caso específico para $N = 3$. El vector vendría dado por la siguiente expresión:

$$v(X) = \frac{1}{3}(-3, -1, -1, 1, -1, 1, 1, 3) \quad (3.5)$$

Por ejemplo, la tercera componente del vector $v(X)$, -1 , correspondiente al estado $s = (010) = 2$, se ha calculado como $\frac{1}{3}(-1 + 1 - 1) = \frac{1}{3}(-1) = -\frac{1}{3}$. Esto implica que el hecho de que el conjunto pase de un estado $s(t)$ al estado $s(t+1) = (010) = 2$ (suponiendo que es posible pasar del estado $s(t)$ al $s(t+1)$) supone una pérdida esperada del capital global del conjunto de $-\frac{1}{3}$. Las demás componentes se calculan de forma análoga.

Podemos visualizar la cadena de Markov del juego B'' mediante el diagrama de la figura 3.5:

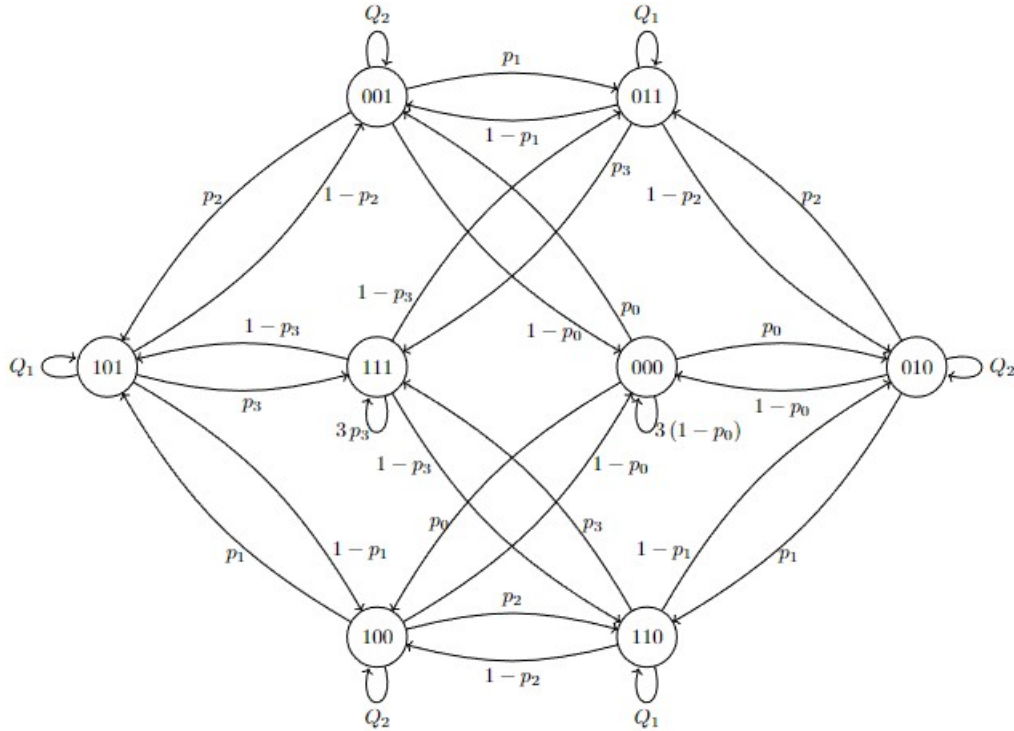


Figura 3.5: Cadena de Markov del juego B'' . Elaboración propia.

En este diagrama, cada una de las probabilidades de transición debería estar multiplicada por $1/3$, lo que no se ha hecho en el propio diagrama para facilitar su lectura.

Podemos observar que esta cadena de Markov es finita y además es ergódica, luego por los teoremas 1.0.1 y 1.0.2 vistos en los preliminares sabemos que existe una única distribución estacionaria y que esta coincide con la distribución límite.

Análogamente a como ya se hizo en la sección dedicada a los juegos dependientes de la historia, para construir la cadena del juego A solamente será necesario hacer la sustitución $p_0 = p_1 = p_2 = p_3 = p$, luego en esta también existirá una única distribución estacionaria que coincide con la distribución límite, lo mismo que sucederá en la mezcla aleatoria de los juegos, como ya sabemos.

Pasemos ahora a calcular las distribuciones estacionarias de los juegos A , B'' y de la mezcla aleatoria. Hasta donde nosotros sabemos, solo dos publicaciones ([27] y [30]) dan las probabilidades exactas de este juego B'' , p_0, p_1, p_2, p_3 , y un cierto valor de ϵ para que se verifique la paradoja, pero no tienen en cuenta la condición de que el juego B'' sea justo para $\epsilon = 0$.

Por esta razón, nos propusimos parametrizar las probabilidades como $p_0 = a - \epsilon$, $p_1 = b - \epsilon$, $p_2 = c - \epsilon$, $p_3 = d - \epsilon$, dejando libres los parámetros a, b, c, d, ϵ .

Al igual que se hizo en el caso dependiente de la historia, en el que se tomaba $p_2 = p_3$, aquí también tomaremos $b = c$, de manera que la probabilidad que tiene un jugador de ganar el juego B'' no depende de si su vecino de la izquierda es ganador y el de su derecha es perdedor o viceversa, sino simplemente del número de vecinos perdedores. Además, tal y como se hizo al construir los juegos dependientes del capital y los dependientes de la historia, fijaremos los parámetros a, b, d de manera que el juego B'' sea justo con $\epsilon = 0$ mientras que la mezcla aleatoria de A y B'' resulte en un juego ganador.

Observación 3.2.1. Si queremos que el juego B'' efectivamente distinga si el vecino perdedor de un jugador es el de su izquierda o el de su derecha, podemos realizarlo implementando ligeras modificaciones en el código del anexo 7.1.13.

Haciendo los cálculos con *Mathematica* como aparece en 7.2.3 con la matriz del juego A , y tomando como ya sabemos $p = \frac{1}{2} - \epsilon$ obtenemos la siguiente distribución estacionaria:

$$\left(\frac{1}{8} + \frac{3\epsilon}{4} + \frac{3\epsilon^2}{2}, \frac{1}{8} + \frac{\epsilon}{4} - \frac{\epsilon^2}{2}, \frac{1}{8} + \frac{\epsilon}{4} - \frac{\epsilon^2}{2}, \frac{1}{8} - \frac{\epsilon}{4} - \frac{\epsilon^2}{2}, \right. \\ \left. \frac{1}{8} + \frac{\epsilon}{4} - \frac{\epsilon^2}{2}, \frac{1}{8} - \frac{\epsilon}{4} - \frac{\epsilon^2}{2}, \frac{1}{8} - \frac{\epsilon}{4} - \frac{\epsilon^2}{2}, \frac{1}{8} - \frac{3\epsilon}{4} + \frac{3\epsilon^2}{2} \right)$$

Como ya ocurría en casos previos, cuando $\epsilon = 0$, todas las componentes de la distribución estacionaria de A son iguales entre sí y, por lo tanto, en este caso valen $\frac{1}{8}$.

A partir de esta distribución estacionaria tenemos que la probabilidad de ganar en el juego A es

$$\frac{1}{2} - \epsilon$$

de modo que el juego A es justo cuando $\epsilon = 0$.

La distribución estacionaria del juego B'' viene dada por un vector demasiado grande, por lo que se ha optado por no incluirlo en este documento y remitimos al código del

anexo 7.2.3. La probabilidad de ganar el juego B'' haciendo un desarrollo de Taylor de orden 1 en el punto $\epsilon = 0$ es:

$$\frac{a + ab - ad}{1 + 2a - b + ab - d - 2ad + bd} + \frac{(-1 - 2a - a^2 - ab - a^2b + b^2 + ab^2 + 2d + 3ad + a^2d - bd - b^2d - d^2 - ad^2 + bd^2)}{(1 + 2a - b + ab - d - 2ad + bd)^2} \epsilon$$

Para la mezcla aleatoria, tanto la distribución estacionaria como la probabilidad de ganar vienen dadas por expresiones demasiado grandes, que remitimos de nuevo al anexo 7.2.3. Sin embargo, como vamos a hacer que ϵ tome el valor 0, nos basta con quedarnos con el primer sumando del desarrollo de Taylor de la probabilidad de ganar la mezcla aleatoria, que es:

$$\frac{(1 + 2a)(2 + b - c)}{8 + a(7 + 2b - 4c) + 2b(-1 + c) - 5c}$$

A partir de las expresiones de las probabilidades de ganar en B'' y en la mezcla aleatoria, podemos estudiar cuándo se da la paradoja de Parrondo, siendo para $\epsilon = 0$ el juego B'' un juego justo.

Los respectivos cálculos en *Mathematica* del anexo 7.2.4 llevan a que el juego B'' es justo y que la mezcla aleatoria tiene una probabilidad mayor que 0.5, para $\epsilon = 0$, cuando se verifican estas condiciones:

a	b	d
$0 < a < 0.5$	$0.5 < b < 1 - a$	$d = \frac{1-b-ab}{1-b}$
$0.5 < a \leq 1$	$1 - a < b < 0.5$	$d = \frac{1-b-ab}{1-b}$

Tabla 3.1: Condiciones de a , b y d para que se dé la paradoja en los juegos cooperativos verificando que B'' es justo para $\epsilon = 0$

Por ejemplo, tomemos $a = \frac{1}{4}$, $b = \frac{6}{10}$, con lo que $d = \frac{5}{8}$. Bajo estas condiciones y con $p = \frac{1}{2} - \epsilon$, $p_0 = a - \epsilon$, $p_1 = b - \epsilon = p_2$, $p_3 = d - \epsilon$, efectivamente los juegos A y B'' son justos para $\epsilon = 0$ y perdedores para $\epsilon > 0$, siendo la mezcla aleatoria de ambos un juego ganador para una determinada región de valores de ϵ .

Estudiemos dicha región. Con los valores de los parámetros dados arriba, la probabilidad de ganar en la mezcla aleatoria utilizando un desarrollo de Taylor de orden 1 en $\epsilon = 0$ es

$$0.50641 - 1.23789\epsilon$$

Por lo tanto, ϵ puede tomar los valores del intervalo $[0, 0.00517241]$ para que la paradoja de Parrondo se verifique.

Cabe destacar que las tablas, condiciones y expresiones mostradas en esta sección no aparecen en la literatura estudiada sobre el tema, razón por la cual nos ha parecido de gran utilidad incluirlas aquí.

Observación 3.2.2. En [12] se muestra que la secuencia de juegos dada por $ABABB$ es la secuencia óptima no superior a 12 turnos en los juegos cooperativos, es decir, la que mayores ganancias proporciona en el estado estacionario de todas las posibles.

3.2.2. Segunda variante

Podemos modificar el juego A de manera que si el jugador i gana el juego, un jugador j de sus vecinos le paga una unidad de capital. En caso contrario, i se la paga a j . Veamos cómo se realiza esta modificación sobre el juego.

Por simplificar cálculos, podemos volver a suponer que $N = 3$, y de nuevo tenemos que todo estado dentro de los 2^3 estados posibles del conjunto lleva asociado un número en base decimal.

Como A es un juego entre dos individuos que son vecinos, sólo tenemos 3 casos que considerar: que el individuo 1 y el 2 compitan, que lo hagan el 2 y el 3 y que lo hagan el 3 y el 1. Supondremos que la probabilidad de que cada jugador gane el juego es la misma para todos, luego cada jugador tiene una probabilidad de $\frac{1}{2}$ de ganar al enfrentarse a otro jugador. Por ejemplo, para pasar del estado 0, (000), al estado 2, (010), el individuo 1 y el individuo 2 deben competir y tiene que ganar el 2. Como la probabilidad de que se dé este caso es de $\frac{1}{3}$, la probabilidad de todo ello es de $\frac{1}{6}$. También podemos conseguir este resultado si el individuo 2 y el individuo 3 compiten, y gana el 2, lo que sucede también con una probabilidad de $\frac{1}{6}$. Por lo tanto, $p_{02} = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$.

De igual manera, para pasar del estado 1, (001), al estado 5, (101), el individuo 1 y el individuo 2 deben competir y tiene que ganar el 1. Por lo tanto, $p_{15} = \frac{1}{6}$. Los demás elementos de la matriz se calculan de manera análoga. Finalmente, obtenemos:

$$\mathcal{P}^A = \begin{pmatrix} 0 & 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 1/3 & 1/6 & 1/6 & 1/6 & 1/6 & 0 & 0 \\ 0 & 1/6 & 1/3 & 1/6 & 1/6 & 0 & 1/6 & 0 \\ 0 & 1/6 & 1/6 & 1/3 & 0 & 1/6 & 1/6 & 0 \\ 0 & 1/6 & 1/6 & 0 & 1/3 & 1/6 & 1/6 & 0 \\ 0 & 1/6 & 0 & 1/6 & 1/6 & 1/3 & 1/6 & 0 \\ 0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 1/3 & 1/3 & 0 \end{pmatrix}$$

La cadena de Markov asociada a este juego es la que podemos observar en la figura 3.6:

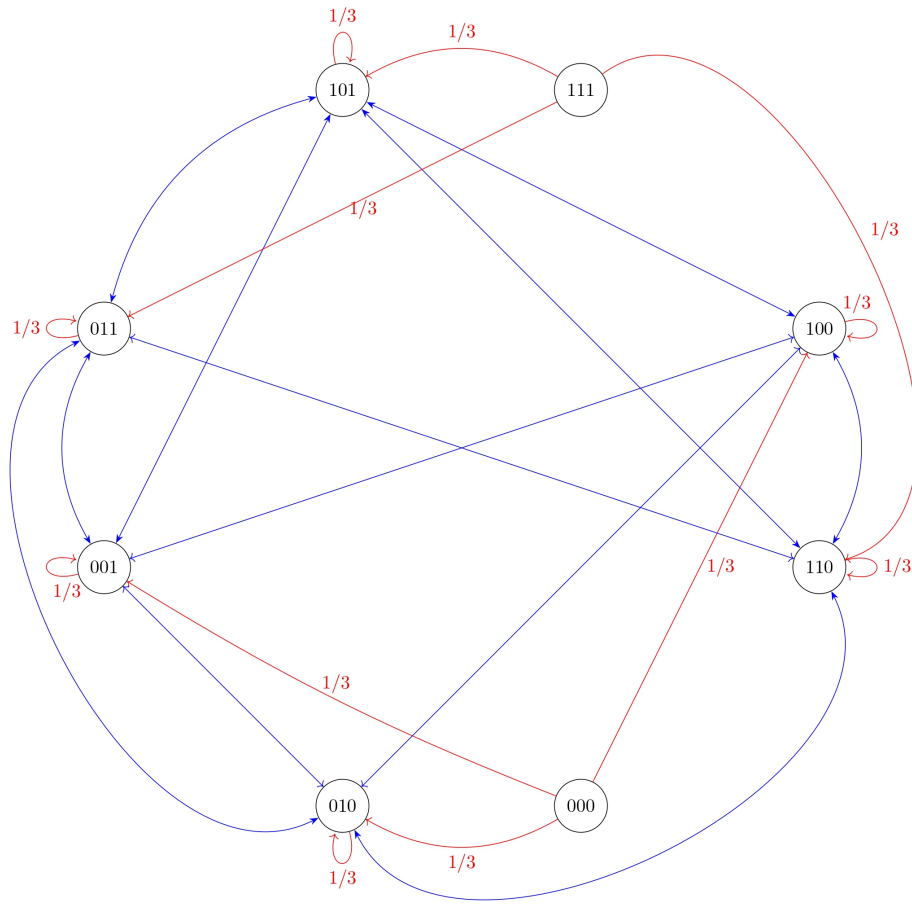


Figura 3.6: Cadena de Markov del juego A modificado. Elaboración propia.

En esta cadena de Markov se ha optado por emplear las flechas bidireccionales azules para representar aquellas transiciones con probabilidad $\frac{1}{6}$ para facilitar la lectura de la figura. Como estas flechas son bidireccionales, permiten pasar de un estado al otro en ambos sentidos.

Calculando la distribución estacionaria asociada a esta matriz con el código en 7.2.5, obtenemos

$$\pi^A = (0, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 0)$$

Este es un resultado lógico, pues a los estados 0 y 7 no se puede llegar desde ningún otro estado, y la probabilidad se distribuye de manera equitativa entre los seis estados restantes.

Además, el juego A definido de esta manera es un juego de suma 0, en el sentido de que el capital total de los jugadores queda inalterado, por lo que también lo podemos entender como un juego justo ya que el capital total de los jugadores ni crecerá ni disminuirá al jugar A .

En este caso, no nos ha parecido necesario realizar un análisis exhaustivo de la probabilidad de ganar en la mezcla aleatoria con el juego B'' , aunque esta podría ser calculada de manera análoga a como se hizo con la primera variante de los juegos cooperativos realizando las modificaciones pertinentes en el código 7.2.3. Pese a no hacer dicho análisis, sí nos ha parecido importante incluir esta variante debido a su importancia en algunos artículos y por el hecho de que más adelante, específicamente en el capítulo 5, retomaremos este enfoque del juego A .

Capítulo 4

Nuevos enfoques

Hasta ahora hemos presentado una extensa revisión de la paradoja de Parrondo y de los diversos casos en los que se presenta. Hemos replicado algunos de los principales resultados y los hemos analizado desde una nueva perspectiva.

En este capítulo, nuestro objetivo será aportar nuevas ideas y resultados al tema para intentar así encontrar situaciones innovadoras en las que la paradoja de Parrondo pueda tener lugar y que hasta ahora apenas se hayan estudiado.

En la primera sección de este capítulo se estudia la existencia de la paradoja en función del parámetro de aleatorización γ , de manera que se estudia sobre cualquier posible combinación aleatoria de los juegos A y B , ya sea en el caso dependiente del capital, en el caso dependiente de la historia o en el caso colectivo. En la segunda sección nos centramos en los juegos de Parrondo originales A y B y estudiamos el espacio de probabilidades $\{p, p_m, p_b\}$ donde la paradoja tiene lugar, continuando de esta manera los estudios realizados en artículos como [18]. Finalmente, en la tercera sección estudiamos la posible existencia de la paradoja de Parrondo al considerar no solamente dos juegos, como se ha hecho hasta ahora, sino tres. Para ello, consideraremos la mezcla aleatoria del juego A original, del juego dependiente del capital y del juego dependiente de la historia.

4.1. Estudio del parámetro γ

A lo largo de la redacción de este trabajo, nos llamó la atención la no existencia de estudios, hasta donde nosotros sabemos, sobre el parámetro γ en la paradoja. Por esta razón en esta sección nos centraremos en el mismo, es decir, no supondremos que tiene un valor fijo de $\frac{1}{2}$ sino que permitiremos cualquier combinación aleatoria de los juegos, esto es, γ puede tomar cualquier valor en el intervalo $[0, 1]$. Esto nos permitirá modelizar situaciones en las que el juego B tiene más probabilidad de ser elegido que el juego A , o viceversa.

Por ejemplo, centrémonos en el primer caso de la paradoja de Parrondo, el dependiente del capital, y estudiemos, para las probabilidades ya conocidas en este caso, para qué valores de γ se produce la paradoja.

De nuevo empleando *Mathematica* se ha calculado la probabilidad de ganar en la combinación aleatoria en función de los parámetros γ y ϵ , que no incluimos aquí debido a su longitud pero puede encontrarse en el anexo 7.2.6.

A partir de esta probabilidad, hemos construido la figura 4.1, en la que vemos cómo la región del plano de coordenadas γ y ϵ donde se puede dar la paradoja viene determinada por una forma parabólica. Observamos que, para $\gamma = 0$ y $\gamma = 1$, el parámetro ϵ únicamente puede tomar el valor 0, lo cual es lógico puesto que en el primer caso se estaría jugando únicamente el juego A y en el segundo únicamente el B , y ambos son juegos justos como ya sabemos.

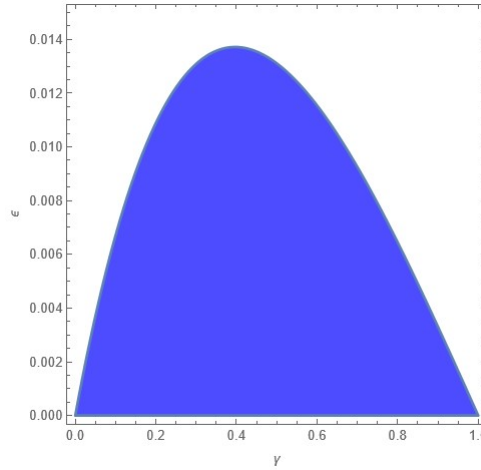


Figura 4.1: Relación entre los parámetros γ y ϵ para los juegos originales. Elaboración propia.

De esta manera, la región en esta figura representa las parejas de puntos (γ, ϵ) donde la combinación aleatoria de A y B es un juego ganador.

El mayor rango de valores que puede tomar el parámetro ϵ se da en el caso en el que γ vale 0.396737, donde ϵ puede llegar a alcanzar el valor 0.0137176.

Sin embargo, este no tiene por qué ser necesariamente el punto donde la probabilidad de ganar en la mezcla aleatoria alcance su máximo. Realizando los cálculos oportunos, se observa que el valor de γ donde la probabilidad alcanza el valor máximo es de 0.414589, siendo 0.513101 el valor máximo de la probabilidad, lógicamente para $\epsilon = 0$.

Podemos hacer también este estudio para los juegos dependientes de la historia. En el anexo 7.2.7 incluimos el cálculo de la probabilidad de ganar en la combinación aleatoria en función de los parámetros γ y ϵ .

De nuevo podemos hacer el gráfico que se observa en la figura 4.2 para ver la región del plano de coordenadas γ y ϵ en la que se da la paradoja:

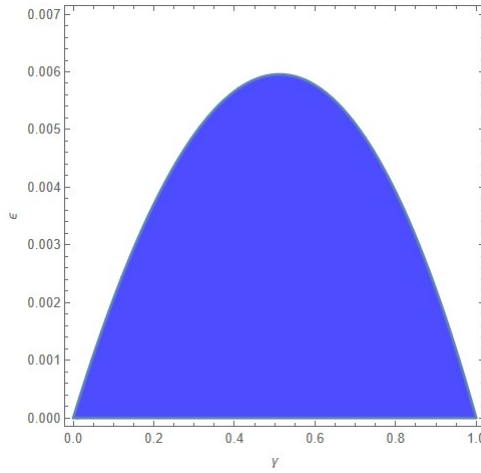


Figura 4.2: Relación entre los parámetros γ y ϵ para el juego dependiente de la historia. Elaboración propia.

El mayor rango de valores de ϵ se alcanza cuando γ vale 0.511911, donde ϵ puede alcanzar el valor 0.00595576. El valor máximo de la probabilidad de ganar se alcanza en 0.498744, y es 0.505828. Observamos entonces que tanto el rango de valores posibles de ϵ como el valor máximo de la probabilidad en la combinación aleatoria son en este caso menores que en el caso dependiente del capital.

Procedemos a estudiar ahora el caso de los juegos cooperativos. Para ello se han empleado las probabilidades elegidas en la sección 3.2.1, esto es, $a = \frac{1}{4}, b = \frac{6}{10} = c, d = \frac{5}{8}$. En este caso, la probabilidad de ganar viene dada por una expresión que podemos encontrar en el código de *Mathematica* en el anexo 7.2.8 y que nos permite calcular el gráfico de la región del plano de coordenadas γ y ϵ en el que se da la paradoja, que podemos observar en la figura 4.3:

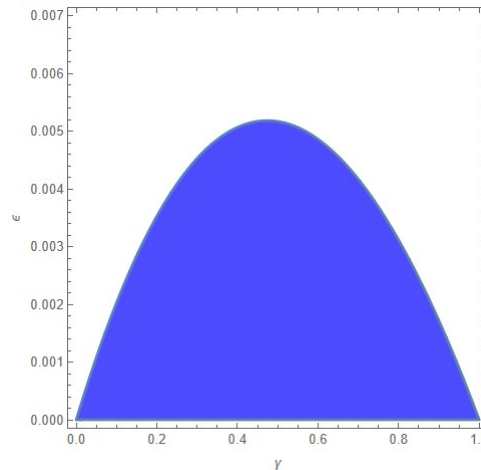


Figura 4.3: Relación entre los parámetros γ y ϵ para el juego cooperativo. Elaboración propia.

El mayor rango de valores de ϵ se alcanza cuando γ vale 0.474069, donde ϵ llega a

alcanzar el valor 0.00518633. El valor máximo de la probabilidad se da cuando γ toma el valor 0.411147. En ese punto, la probabilidad de ganar es 0.506616. En este caso, observamos valores similares a los alcanzados en el caso dependiente de la historia.

Ahora estudiaremos cómo varía la probabilidad de ganar la combinación aleatoria de los juegos en función del parámetro γ , para un cierto valor de ϵ prefijado. Para ello, hemos construido el gráfico de la figura 4.4 en *Mathematica*, para la combinación aleatoria de los juegos originales A y B de la paradoja de Parrondo:

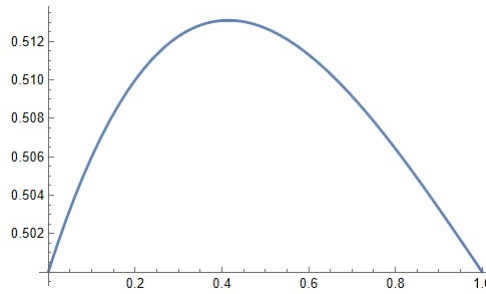


Figura 4.4: Relación entre la probabilidad de ganar de la combinación aleatoria de los juegos originales A y B y el parámetro γ . Elaboración propia.

Como es lógico, para los valores de $\gamma = 0$ y de $\gamma = 1$, correspondientes a solamente jugar el juego B o el A , respectivamente, el valor de la probabilidad es 0.5, pues dicha imagen se corresponde con el caso $\epsilon = 0$.

En este caso, como ya se mencionó previamente, se observa que el valor de γ que maximiza la probabilidad de ganar es 0.414589, tomando dicha probabilidad el valor 0.513101. Pese a que este es el valor que maximiza la probabilidad, podemos observar cómo para cualquier posible valor de γ la paradoja se sigue dando, puesto que los valores son siempre mayores o iguales que 0.5.

Un estudio análogo podría realizarse sobre el caso dependiente del capital y el caso cooperativo sin más que realizar las modificaciones oportunas en el código. Cabe mencionar que a lo largo del anexo dedicado a esta sección se presenta el código de diversas salidas interactivas que permiten al usuario modificar los parámetros γ y ϵ a conveniencia en muchos de los resultados presentados aquí, como en el cálculo de la probabilidad de ganar, el gráfico de esta o el valor de γ en el que esta alcanza su máximo y el valor de dicho máximo.

Con esto, hemos visto que la paradoja de Parrondo se da no solamente en el caso de la mezcla aleatoria de los juegos, es decir, con $\gamma = \frac{1}{2}$, sino en una amplia variedad de casos donde γ puede tomar cualquier valor en el intervalo $[0, 1]$.

4.2. Estudio del espacio de parámetros

En esta sección consideraremos el espacio tridimensional $\mathbb{R}^3$ dado por las probabilidades $\{p, p_m, p_b\}$ empleadas en los juegos de Parrondo originales. De este modo, seremos capaces de ver en qué puntos de $\mathbb{R}^3$ se da la paradoja de Parrondo, ahora sin depender del parámetro ϵ con el que hemos venido trabajando. Para ello hemos creado un nuevo código en 7.2.9.

Construyendo la matriz del juego B sin especificar las probabilidades, es decir, escribiéndola en la forma

$$\mathcal{P}_B = \begin{pmatrix} 0 & p_m & 1 - p_m \\ 1 - p_b & 0 & p_b \\ p_b & 1 - p_b & 0 \end{pmatrix}$$

se tiene que la distribución estacionaria, una vez normalizada, es:

$$\begin{aligned} \pi_0 &= - \left(\frac{-1 + p_b - p_b^2}{(1 - p_m + p_b p_m) \left(1 - \frac{-1 + p_b - p_b^2}{1 - p_m + p_b p_m} - \frac{-1 + p_b - p_b p_m}{1 - p_m + p_b p_m} \right)} \right) \\ \pi_1 &= - \left(\frac{-1 + p_b - p_b p_m}{(1 - p_m + p_b p_m) \left(1 - \frac{-1 + p_b - p_b^2}{1 - p_m + p_b p_m} - \frac{-1 + p_b - p_b p_m}{1 - p_m + p_b p_m} \right)} \right) \\ \pi_2 &= \frac{1}{1 - \frac{-1 + p_b - p_b^2}{1 - p_m + p_b p_m} - \frac{-1 + p_b - p_b p_m}{1 - p_m + p_b p_m}} \end{aligned}$$

Y la probabilidad de ganar el juego es:

$$\begin{aligned} & - \frac{(-1 + p_b - p_b^2)p_m}{(1 - p_m + p_b p_m) \left(1 - \frac{-1 + p_b - p_b^2}{1 - p_m + p_b p_m} - \frac{-1 + p_b - p_b p_m}{1 - p_m + p_b p_m} \right)} + \\ & + p_b \left(1 + \frac{-1 + p_b - p_b^2}{(1 - p_m + p_b p_m) \left(1 - \frac{-1 + p_b - p_b^2}{1 - p_m + p_b p_m} - \frac{-1 + p_b - p_b p_m}{1 - p_m + p_b p_m} \right)} \right) \end{aligned}$$

A partir de esta expresión, la región del plano donde el juego B es perdedor viene representada en la figura 4.5, donde en el eje X se representa la variable p_m y en el eje Y la variable p_b :

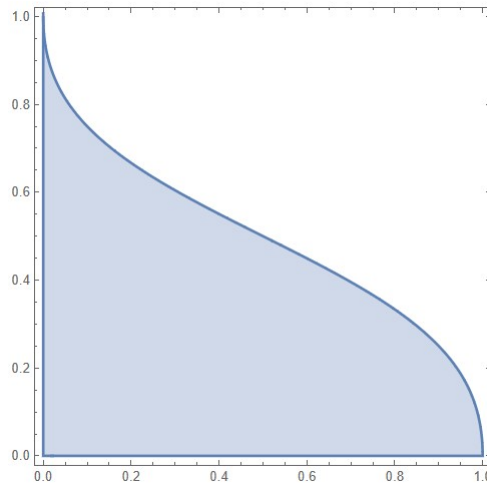


Figura 4.5: Región donde el juego B es perdedor. Elaboración propia.

A partir de este gráfico, la región del espacio de parámetros $\{p, p_m, p_b\}$ donde los juegos A y B son perdedores se presenta en la figura 4.6:

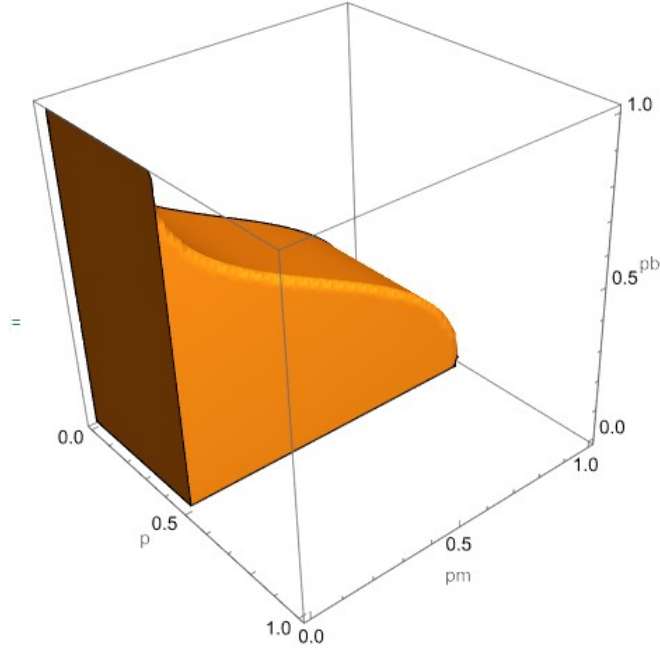


Figura 4.6: Región donde los juegos A y B son perdedores. Elaboración propia.

Lógicamente, este gráfico se construye a partir del anterior, considerando los tríos de puntos (p, p_m, p_b) tal que $p \leq 0.5$ y tal que (p_m, p_b) ya pertenecían al gráfico anterior.

Finalmente, la región del espacio de parámetros donde se da la paradoja es la de la figura 4.7:

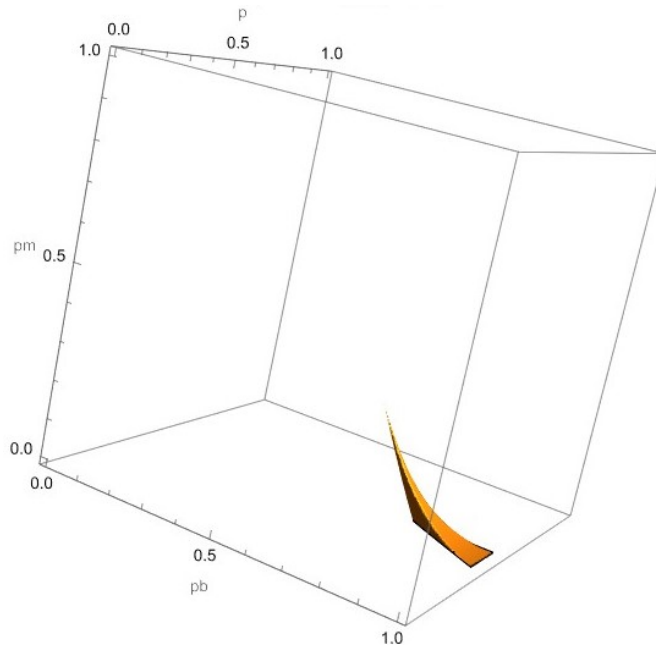


Figura 4.7: Región donde se da la paradoja de Parrondo. Elaboración propia.

Por lo tanto, en esta región de puntos se verifica que los juegos A y B son perdedores, mientras que su mezcla aleatoria es ganadora.

Observamos que tan solo en una pequeña región del espacio tridimensional se llega a dar la paradoja de Parrondo. Así, hemos podido visualizar el conjunto de puntos tridimensionales dados por las probabilidades donde la paradoja de Parrondo se da en los juegos originales.

En el anexo de código dedicado a esta sección, se muestra un gráfico interactivo que, aunque es computacionalmente costoso, muestra cómo evoluciona el espacio de probabilidades tridimensional mostrado anteriormente según el parámetro γ de la combinación aleatoria de A y B varía.

Observación 4.2.1. Para realizar este estudio en los juegos dependientes de la historia y en los juegos cooperativos, sería necesario fijar dos de las cinco probabilidades con las que se trabaja en estos tipos de juegos de manera que podamos visualizar un espacio tridimensional como se ha hecho aquí.

4.3. ¿Existe la paradoja de Parrondo sobre tres juegos?

A lo largo de la lectura de los artículos empleados para realizar este trabajo, observamos que prácticamente ninguno de ellos se centra en la posible existencia de la paradoja de Parrondo en tres o más juegos. Ante esta situación, nos planteamos la siguiente cuestión: dados tres juegos A , B y C , siendo los tres justos o perdedores, ¿será la mezcla aleatoria de ellos un juego ganador?

Para responder a esta pregunta, elegimos como juego A al primer juego presentado en este trabajo, con una probabilidad p de ganar y $1 - p$ de perder. Como segundo juego, elegimos el juego dependiente del capital, que funciona de la manera conocida según si este es o no un múltiplo de 3, y como tercer juego el juego dependiente de la historia.

Para poder tener en cuenta los tres juegos a la vez es necesario trabajar ahora con doce estados posibles, que son:

$$\{(0, LL), (1, LL), (2, LL), (0, LW), (1, LW), (2, LW), \\ (0, WL), (1, WL), (2, WL), (0, WW), (1, WW), (2, WW)\}$$

La notación empleada en estos estados es la análoga a la empleada en el capítulo previo: 0,1,2 para el capital del jugador módulo 3, L y W para el resultado de sus dos juegos anteriores, siendo L una derrota, es decir, una pérdida de una unidad en el capital del jugador, y W una victoria. Trabajando con estos estados, tenemos en cuenta a la vez las características de los juegos B y C .

La cadena de Markov que define el juego A es la siguiente:

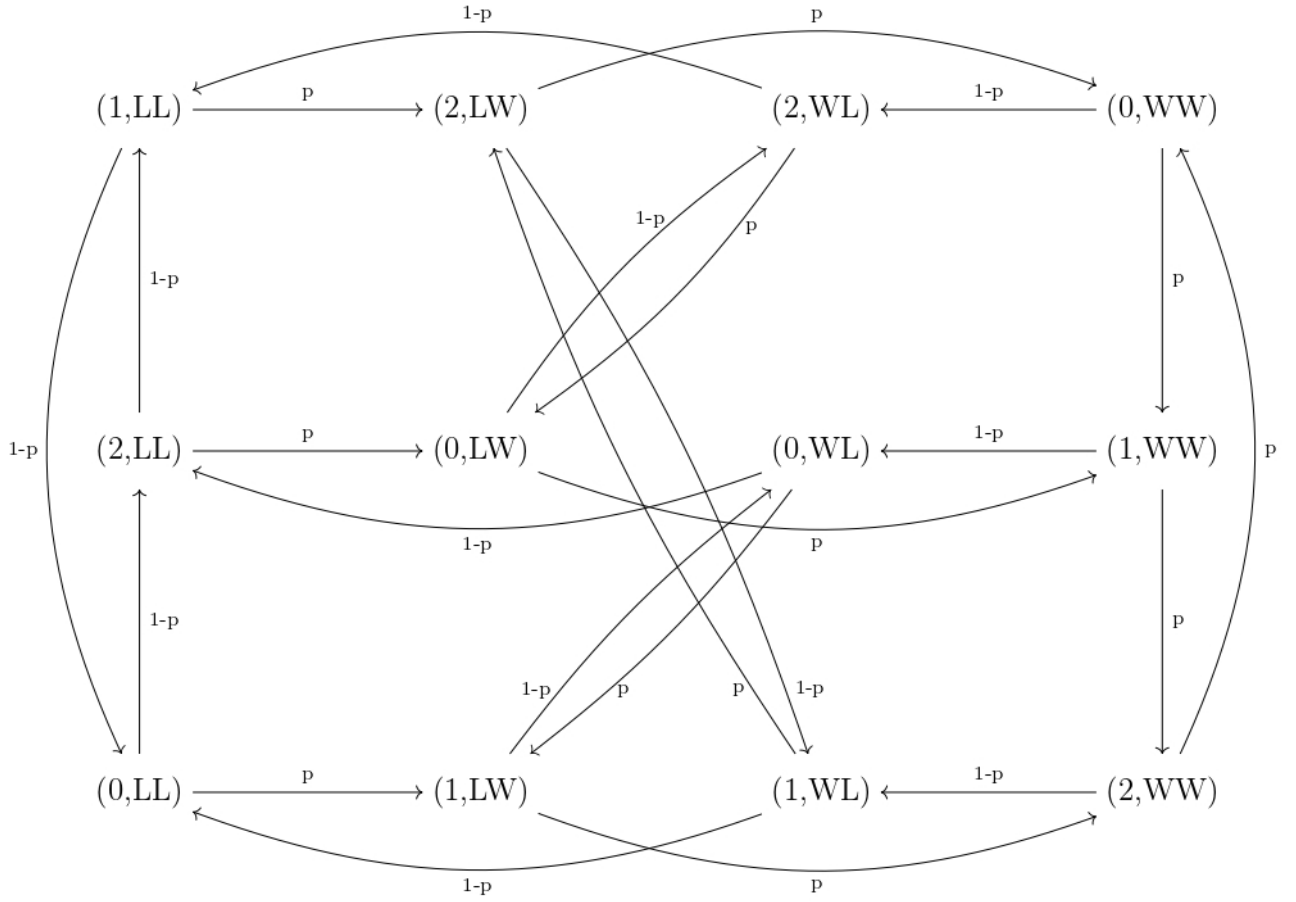


Figura 4.8: Cadena de Markov del juego A al considerar la mezcla aleatoria de A , B y C . Elaboración propia.

Para construir la cadena se han calculado todas las probabilidades de transición entre un estado y otro. Por ejemplo, si un jugador en el estado $(1, LL)$ gana A , lo cual sucede con probabilidad p , su capital aumenta en una unidad y su historia pasará de LL a LW . Sin embargo, si en ese estado pierde A , que sucede con probabilidad $1 - p$, su capital será 0 y su historia seguirá siendo LL .

Observamos que es una cadena de Markov con un número de estados finitos y donde todos ellos intercomunican, son recurrentes positivos y aperiódicos, luego es una cadena de Markov ergódica y por lo tanto podemos aplicar los teoremas de los preliminares 1.0.1 y 1.0.2. Es decir, la cadena de Markov tiene una única distribución estacionaria que además coincide con la límite.

Como el juego A no depende del estado en el que nos encontremos, las únicas probabilidades de transición que aparecen son p y $1 - p$ para ganar o perder respectivamente. La cadena de Markov del juego B se podría construir a partir de esta sin más que sustituir p por p_m o p_b en función de si en ese estado el capital es múltiplo o no de 3. Análogamente haríamos para la cadena de Markov del juego C empleando las probabilidades p_1, p_2, p_3, p_4 donde correspondiesen según los estados, es decir, p_1 para un estado del tipo LL , p_2 para un estado del tipo LW , p_3 para un estado del tipo WL y p_4 para un estado del tipo WW . Por lo tanto, los teoremas siguen siendo válidos para estas cadenas, y por lo tanto también para su mezcla aleatoria.

Para ejemplificar más claramente esto, tenemos que la matriz de transición del juego A es la siguiente:

$$A = \begin{pmatrix} 0 & 0 & 1-p & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p & 0 & 0 & 0 & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & p & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & 0 & p \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & p & 0 & 0 \\ 0 & 0 & 1-p & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p & 0 & 0 & 0 & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & p & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & 0 & p \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & p & 0 & 0 \end{pmatrix}$$

En ella, se ha respetado el orden presentado en el conjunto de estados, es decir, la primera fila y la primera columna son las asociadas al estado $(0, LL)$, mientras que la última fila y columna son las asociadas a $(2, WW)$.

La matriz de transición del juego B es:

$$B = \begin{pmatrix} 0 & 0 & 1-p_m & 0 & p_m & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p_b & 0 & 0 & 0 & 0 & p_b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p_b & 0 & p_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p_m & 0 & p_m & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p_b & 0 & 0 & 0 & 0 & p_b \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p_b & 0 & p_b & 0 & 0 \\ 0 & 0 & 1-p_m & 0 & p_m & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p_b & 0 & 0 & 0 & 0 & p_b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p_b & 0 & p_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p_m & 0 & p_m & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p_b & 0 & 0 & 0 & 0 & p_b \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p_b & 0 & p_b & 0 & 0 \end{pmatrix}$$

Y la del juego C es:

$$C = \begin{pmatrix} 0 & 0 & 1-p_1 & 0 & p_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p_1 & 0 & 0 & 0 & 0 & 0 & p_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p_1 & 0 & p_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p_2 & 0 & p_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p_2 & 0 & 0 & 0 & 0 & p_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p_2 & 0 & p_2 & 0 & 0 \\ 0 & 0 & 1-p_3 & 0 & p_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p_3 & 0 & 0 & 0 & 0 & 0 & p_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p_3 & 0 & p_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p_4 & 0 & p_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p_4 & 0 & 0 & 0 & 0 & p_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p_4 & 0 & p_4 & 0 & 0 \end{pmatrix}$$

A partir de estas tres matrices, podemos calcular la matriz de la mezcla aleatoria de la forma en que ya sabemos, esto es:

$$\frac{1}{3}A + \frac{1}{3}B + \frac{1}{3}C$$

Una vez construidas las cuatro matrices de transición, estamos en disposición de calcular las probabilidades de ganar, una vez los juegos entran en su fase estacionaria. Para ello, emplearemos las probabilidades con las que ya se trabajó previamente, es decir, $p = \frac{1}{2} - \epsilon$, $p_m = \frac{1}{10} - \epsilon$, $p_b = \frac{3}{4} - \epsilon$, $p_1 = \frac{9}{10} - \epsilon$, $p_2 = \frac{1}{4} - \epsilon = p_3$ y $p_4 = \frac{7}{10} - \epsilon$. A través de los cálculos realizados con *Mathematica* presentes en el código del anexo 7.2.10, se obtiene que la probabilidad de ganar el juego *A* haciendo un desarrollo de Taylor de orden 2 en $\epsilon = 0$ es:

$$\frac{1}{2} - \epsilon$$

Es importante hacer ver que, como podemos deducir de casos anteriores, la distribución estacionaria del juego *A* en este caso sigue siendo equiprobable en todos sus estados, es decir, viene dada por un vector de doce componentes donde cada una de las componentes tiene un valor de $\frac{1}{12}$.

Análogamente, la probabilidad de ganar el juego *B* es:

$$\frac{1}{2} - \frac{147}{169}\epsilon + \frac{1704}{28561}\epsilon^2$$

Y la del juego *C* es:

$$\frac{1}{2} - \frac{10}{9}\epsilon + \frac{100}{81}\epsilon^2$$

Por lo tanto, observamos que los tres juegos son justos para $\epsilon = 0$. Finalmente, de nuevo haciendo un desarrollo de Taylor de orden 2 en $\epsilon = 0$ la probabilidad de ganar en la mezcla aleatoria es:

$$0.513256 - 0.96357\epsilon + 0.329533\epsilon^2$$

A partir de estas expresiones podemos concluir que efectivamente sí existe un pequeño rango de valores de ϵ positivos que permiten que los tres juegos de manera individual sean perdedores, pero que su mezcla aleatoria sea ganadora. Haciendo el respectivo cálculo, observamos que dicha región de valores es el intervalo $[0, 0.0138217]$.

Hemos demostrado pues que, para las probabilidades con las que ya se trabajó anteriormente, la paradoja de Parrondo se sigue verificando para la mezcla aleatoria del juego A , el juego dependiente del capital y el juego dependiente de la historia.

Este es un resultado fundamental de este trabajo ya que no conocemos ningún estudio similar en la extensa bibliografía empleada durante su redacción.

Capítulo 5

Aplicaciones

En este capítulo, nuestro objetivo es aplicar la paradoja de Parrondo sobre otros modelos matemáticos. En la primera sección, relacionaremos la paradoja de Parrondo con las redes complejas, empleando para ello una modificación de la segunda variante de los juegos cooperativos de manera que la podamos estudiar sobre redes completas y sobre redes de mundo pequeño. [27] ha sido la fuente en la que nos hemos basado fundamentalmente para su redacción.

En la segunda sección modificaremos la segunda variante de los juegos cooperativos de manera que puedan ser jugados de manera síncrona, es decir, en cada turno no se elige un solo jugador sino que todos los jugadores juegan a la vez. A partir de aquí, relacionaremos este modelo con algunos sistemas de votación.

Finalmente, en la tercera sección hablaremos sobre algunas aplicaciones importantes que, sin desarrollarlas a fondo, nos han parecido de suficiente calado como para mencionarlas.

5.1. Aplicación sobre redes complejas

Un aspecto crucial de las dinámicas sociales es el estudio de la cooperación y la competición entre los agentes en un grupo. Para modelizar estas interacciones se pueden utilizar redes complejas, donde cada nodo representa a un agente, y dos agentes interaccionan entre sí si existe una arista entre sus respectivos nodos, de manera que se considera que son nodos vecinos. Específicamente emplearemos redes completas y redes de mundo pequeño. Las primeras son aquellas en las que todos los nodos de la red interaccionan entre sí, es decir, están conectados con los demás, mientras que las redes de mundo pequeño son aquellas en las que cada nodo tiene un número relativamente bajo de vecinos pero sin embargo el camino mínimo entre dos nodos tiene una longitud pequeña. La importancia de las redes de mundo pequeño deriva del hecho de que aparecen en una amplia variedad de situaciones sociales.

De esta manera, el modelo basado en agentes e inspirado en la paradoja de Parrondo es el siguiente, adaptado de [27]: tenemos un juego A que es un juego de suma 0 entre los agentes o jugadores, en el sentido de que las ganancias de un jugador se equilibran con las pérdidas de otro y, por lo tanto, el capital del conjunto de jugadores se mantiene constante. En ese sentido, el juego A puede ser considerado como un juego justo. Además, tenemos un segundo juego B , que es el que ya conocemos en el que se observa si el capital

de cada individuo es o no múltiplo de 3, y que ya sabemos que es un juego perdedor para el capital individual y, con ello, también para el global.

Consideramos un conjunto de N agentes y en cada instante de tiempo un jugador i es elegido aleatoriamente de entre los N . Una vez elegido juega aleatoriamente el juego A con una probabilidad p o el juego B con una probabilidad $1 - p$. Es importante no confundir esta probabilidad p con la usada en capítulos previos, pues en este caso es la probabilidad de jugar A , es decir, cumple el papel que anteriormente tenía γ . Si se juega A , se elige, también de manera aleatoria, a otro jugador $j \neq i$, donde j es uno de los vecinos de i en la red. Si se juega B , entonces estamos en el caso conocido y se observa si el capital de i es múltiplo de M . Podemos observar el resumen de las dinámicas del juego en la figura 5.1:

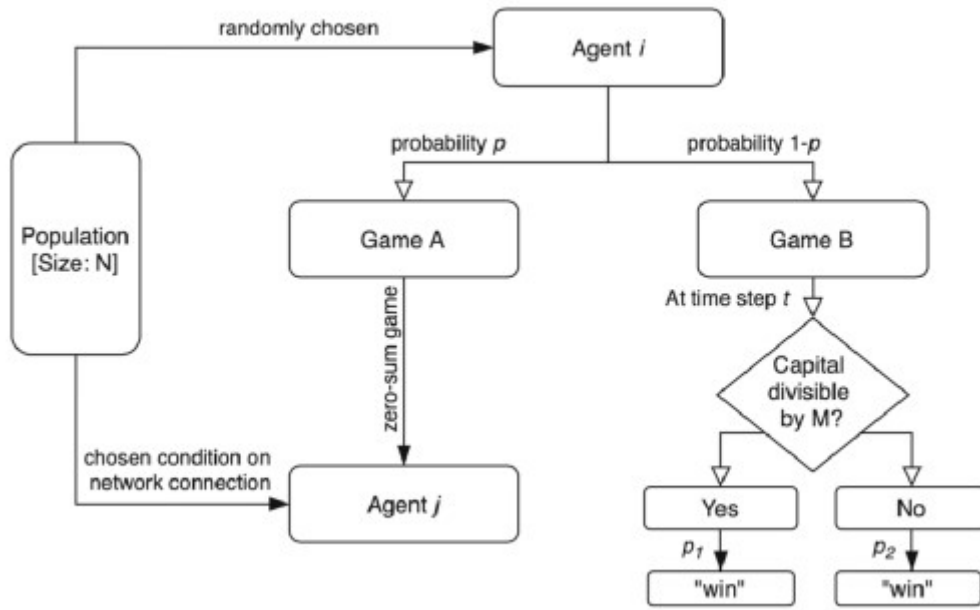


Figura 5.1: Dinámicas del juego sobre redes complejas. Figura extraída de [27].

Todos los agentes comienzan el juego con la misma cantidad de capital inicial C_0 . El juego A no genera ni crecimiento ni disminución en el capital total del conjunto de jugadores. Sin embargo, es clave para que se dé la paradoja, ya que cambia la distribución del capital en el conjunto de jugadores. Las dinámicas del juego A vienen dadas por las cuatro posibles estrategias que pueden seguir sus jugadores:

1. Estrategia de competición: el agente i y el agente j compiten entre sí, cada uno con una probabilidad de ganar de 0.5. Si i gana, j le da una unidad de su capital a i . En caso contrario, es i el que se la paga a j .
2. Estrategia de cooperación: el agente i directamente le da una unidad de capital al agente j .
3. Estrategia basada en la equidad: cuando el capital $C_i(t) \geq C_j(t)$, el agente i le paga una unidad de capital al agente j . En caso contrario, cuando $C_i(t) < C_j(t)$, j paga una unidad de capital a i .
4. Estrategia de competición entre pobres y cooperación entre ricos: cuando el capital

$C_i(t) \geq C_0$, el jugador i decidirá cooperar con el jugador j , es decir, se registrará por el patrón 2. En caso contrario, cuando $C_i(t) < C_0$, se registrará por el patrón 1 y competirá con el jugador j .

De esta manera, la cuarta estrategia es un poco más compleja que las otras tres, y hace que cada jugador se comprometa con el conjunto cuando le va bien pero busque su propio beneficio cuando le va mal, de ahí el nombre que se le ha dado a la estrategia.

Para realizar las simulaciones hemos otorgado el mismo capital inicial $C_0 = 10$ a todos los jugadores, y se han empleado los parámetros $N = 500$ y el número de iteraciones ha sido 20. Además, para hacer que el juego B sea un juego perdedor, se emplean las probabilidades originales que ya conocemos, esto es, $p = 0.5$, $p_m = 0.1 - \epsilon$, $p_b = 0.75 - \epsilon$, donde $\epsilon = 0.005$ y $M = 3$. Los resultados se han promediado sobre 100 simulaciones.

Para no saturar el trabajo con gráficos, se muestran únicamente los dos gráficos referidos a la evolución del capital total a lo largo de 20 iteraciones, para el caso de la red completa y para la red de mundo pequeño, e imponiendo el patrón de comportamiento 4 en todos los jugadores.

Aplicando los códigos de *Python* que aparecen en los anexos 7.1.15 y sucesivos, se puede ver que todos los patrones generan un crecimiento del capital total similar en ambos tipos de redes al de las gráficas 5.2 y 5.3.

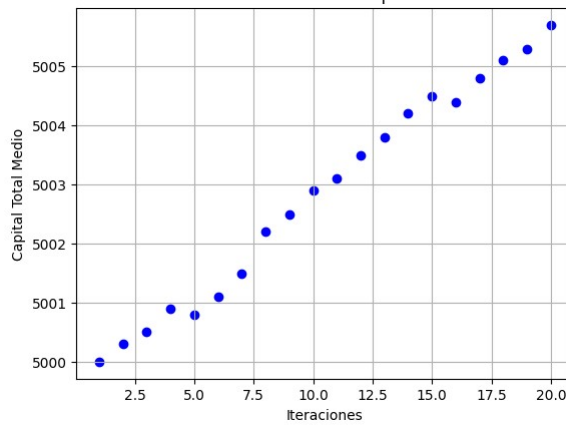


Figura 5.2: Evolución del capital para la red completa con el patrón 4. Elaboración propia.

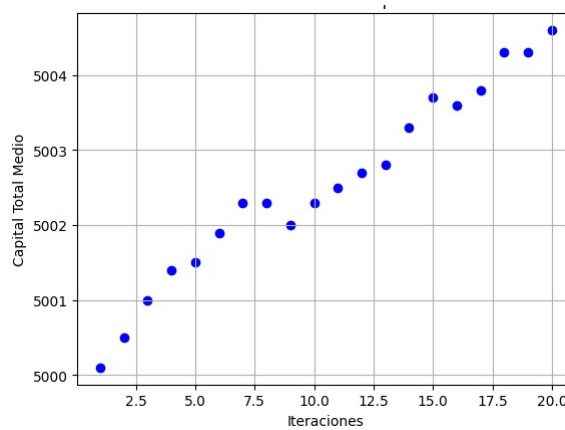


Figura 5.3: Evolución del capital para la red de mundo pequeño con el patrón 4. Elaboración propia.

Por lo tanto, se ha dado la paradoja de Parrondo ya que la combinación aleatoria de dos juegos justos o perdedores para el capital total ha generado un crecimiento del mismo.

A continuación, planteamos un nuevo juego. En él, cada uno de los jugadores puede comportarse de una determinada manera, distinguiendo así entre jugadores “egoístas” y jugadores “altruistas”.

Para elegir el tipo de cada jugador en cada simulación se ha tomado una probabilidad de 0.5 de que pueda ser egoísta. Los jugadores que sean egoístas únicamente jugarán al juego B , de manera que observarán si su capital es o no múltiplo de M y ganarán con probabilidades p_m o p_b , dependiendo del capital individual con el que cuenten en dicha iteración. Los jugadores altruistas, por su parte, juegan el juego A con una probabilidad p y el juego B con una probabilidad $1 - p$. El juego B es el mismo que juegan los jugadores egoístas, mientras que en el juego A se elige un jugador j de entre sus vecinos, y el jugador i directamente le paga una unidad de su capital al jugador j .

De nuevo el juego A es un juego justo respecto al capital global, aunque visto desde el punto de vista del jugador que lo juega es obviamente un juego perdedor para su capital individual. Por su parte, el juego B es un juego perdedor respecto al capital global. Para hacer el análisis del juego se han empleado 100 simulaciones de 20 iteraciones cada una, promediando así los resultados. Además, se ha tomado $N = 50$ y las demás variables se han inicializado como en el caso de las gráficas 5.2 y 5.3. Tras las simulaciones, cuyo código está presente en 7.1.23, en este caso observamos que efectivamente se genera un juego ganador, pues el capital total del conjunto de jugadores crece en la figura 5.4:

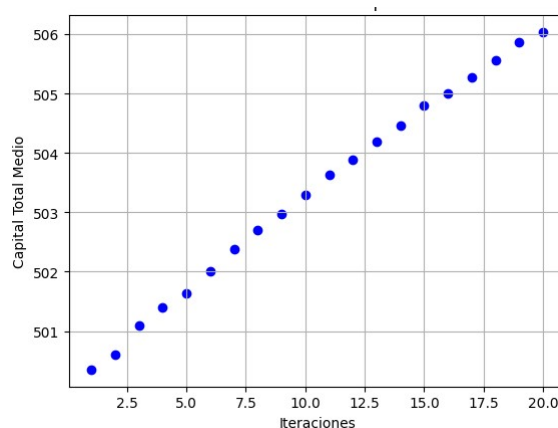


Figura 5.4: Evolución del capital para la red de mundo pequeño separando en jugadores altruistas y egoístas. Elaboración propia.

Sin embargo, haciendo los cálculos en cada uno de los dos subconjuntos de jugadores, observamos que los jugadores egoístas tienden a tener una ganancia individual positiva, de en torno a 0.004 en cada iteración por jugador, mientras que los altruistas van perdiendo dinero con el tiempo, pese a que el capital del conjunto de jugadores sea creciente, ya que su pérdida es de -0.003 por jugador en cada iteración. Es decir, aunque el capital del conjunto crezca, “los pobres se hacen más pobres y los ricos más ricos”. Estos cálculos están presentes en [7.1.24](#).

Como conclusión a esta parte, podemos observar que hemos llegado a un curioso resultado. Los jugadores altruistas, si pudieran elegir si ser o no altruistas, dejarían de serlo ya que comenzarían a ver cómo el ser egoístas les hace ganar más dinero individualmente. Sin embargo, al no haber jugadores altruistas, únicamente se jugaría al juego *B*, que como ya sabemos hace que el capital total decrezca, con lo que en última instancia todos los jugadores acabarían perdiendo. Es el hecho de jugar *A* lo que, pese a ser perjudicial individualmente, es beneficioso a nivel colectivo.

Este caso se ha realizado sobre redes de mundo pequeño ya que nos ha parecido el tipo de red más adecuada sobre la que trabajar, pero se podría aplicar sobre una red completa o sobre cualquier otro tipo de red compleja sin más que realizar los cambios pertinentes en el código.

Observación 5.1.1. Se podría modificar el juego *B* de manera que en vez de ser dependiente del capital dependiera de la historia pasada de cada uno de los jugadores. No lo hemos incluido en este trabajo debido a que los resultados alcanzados con esta situación son similares a los ya mostrados, pero el lector interesado puede investigar más sobre ello en [\[27\]](#).

5.2. Estudio de situaciones síncronas

La toma colectiva de decisiones es un tema importante en áreas tan diversas como la economía, la comunicación, el *machine learning*, la teoría de juegos y la teoría del control, y especialmente en la modelización de la opinión pública. Por ello en esta sección tratamos de relacionar este tema con la paradoja de Parrondo.

Específicamente, estudiaremos el caso síncrono de los juegos cooperativos. Es decir, partiendo del juego original, en el que se elegía aleatoriamente a un jugador de entre los N posibles y este jugaba al juego A o B de manera aleatoria, ahora todos los jugadores jugarán a la vez y serán libres de votar el juego al que quieren jugar en cada turno, aunque imponiendo la condición de que todos deben jugar al mismo juego. En caso de que se diera empate en la votación, el juego elegido será el B .

Hemos considerado tres normas para decidir a qué juego se quiere jugar. En la primera, los jugadores eligen de manera racional el juego de su preferencia y el juego elegido en cada turno será el que más votos haya obtenido.

Podemos deducir fácilmente cuál será el pensamiento de cada uno de los jugadores: si su capital individual es múltiplo de 3, preferirá jugar el juego A , pues su probabilidad de ganar en ese caso es de $\frac{1}{2} - \epsilon$. Sin embargo, si su capital no es múltiplo de 3, preferirá jugar el juego B , ya que entonces su probabilidad de ganar es $\frac{3}{4} - \epsilon$. Empleando *Python* para hacer la simulación con el código 7.1.25 obtenemos el gráfico de la figura 5.5, donde se ha tomado $N = 500$, se han promediado los resultados sobre 100 simulaciones, cada simulación de 100 iteraciones, y el capital inicial de cada jugador es 10, con $\epsilon = 0.005$:

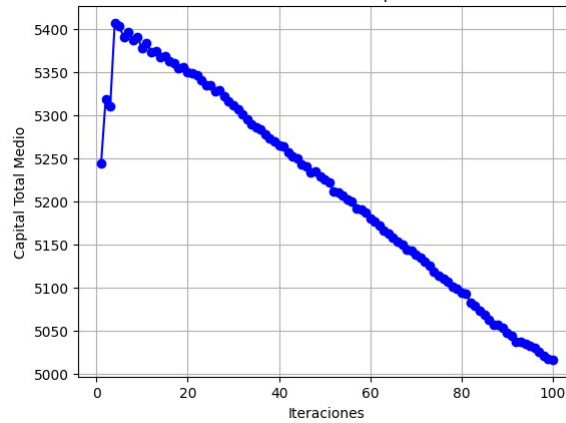


Figura 5.5: Evolución del capital para la situación democrática. Elaboración propia.

Para nuestra sorpresa, observamos que los jugadores, pese a poder elegir de manera individual la mejor opción para ellos, acaban perdiendo dinero como conjunto. Sin embargo, este hecho tiene una fácil explicación. En la gran mayoría de veces (siempre que haya hasta un 50 % de jugadores cuyo capital sea múltiplo de 3), el juego elegido será el B , pues como se mencionó antes es, a primera vista, el juego más favorable desde un punto de vista individual. Como ya sabemos B es un juego perdedor a largo plazo, lo que hace que el capital del conjunto decrezca. Es decir, los jugadores únicamente se fijan en su beneficio a corto plazo, en vez de pensar en el beneficio del grupo a largo plazo.

Observación 5.2.1. El comportamiento a largo plazo de los jugadores es votar siempre B porque el número de personas con un capital individual múltiplo de 3 tiende a la primera componente de la distribución estacionaria de B , esto es, tiende aproximadamente a $\frac{5}{13}$, que es menor que $\frac{1}{2}$. Esto también podemos verlo en el gráfico anterior, donde el capital crece en las primeras iteraciones porque se alterna entre A y B , pero decrece una vez que el único resultado de la votación es B .

Tratando de paliar esa situación, consideramos una segunda norma para elegir a qué juego se quiere jugar. Supongamos que en vez de elegir racionalmente cada jugador elige el juego A o B de manera aleatoria con una probabilidad de 0.5 en cada caso. Dicha situación genera el gráfico 5.6:

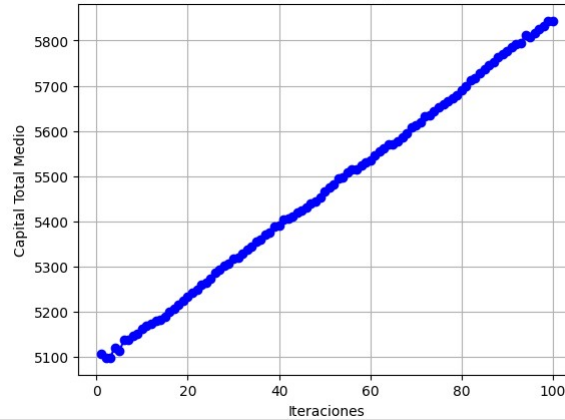


Figura 5.6: Evolución del capital para la situación aleatoria. Elaboración propia.

Para generar dicho gráfico, se ha empleado 7.1.26. Ahora el capital del conjunto sí es creciente y el conjunto de los jugadores sí comienza a ganar dinero, ya que hay jugadores que han votado A incluso cuando tenían más probabilidades de ganar jugando B , lo que ha hecho que aparezca la paradoja de Parrondo. Con ello, el capital individual de cada uno de los jugadores crece también.

Este hecho nos puede hacer reflexionar sobre cómo a veces buscar únicamente el bienestar individual puede no ser el mejor camino, sino que puede ser mejor opción buscar el bien común del grupo y, con ello, también el personal.

Observación 5.2.2. Para que la situación mostrada aquí tenga lugar es necesario elegir un número alto de jugadores. Por ejemplo, es fácil ver que en el caso trivial $N = 1$, la estrategia en la que el jugador decide racionalmente el juego es mejor que la aleatoria, no así a medida que N va creciendo.

Finalmente, proponemos un tercer comportamiento. En este, un jugador elegido al azar al comienzo de la simulación tomará el rol de “dictador”, de manera que todos los demás jugarán al juego que él decida. Empleando los mismos parámetros que en las simulaciones anteriores obtenemos la gráfica 5.7:

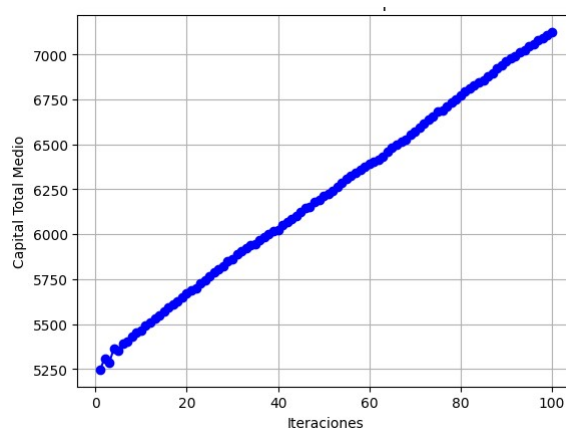


Figura 5.7: Evolución del capital para la situación del dictador. Elaboración propia.

Para la elaboración del gráfico hemos empleado el código 7.1.27. Podemos observar que, de nuevo, el capital del conjunto crece. Es más, el hecho de que haya un jugador decidiendo por todo el grupo hace que el capital del conjunto crezca aún más que en el caso aleatorio. De esta manera, aunque obviamente el capital individual del dictador crece en mayor medida que el de los demás, el capital del conjunto crece de manera muy positiva también.

Observación 5.2.3. Tal y como se hace en [35], de nuevo el juego B podría ser modificado para no depender del capital de cada jugador sino de su historia. Bajo esta modificación, se sigue verificando que la estrategia aleatoria es mejor que la estrategia democrática.

5.3. Otras aplicaciones

La paradoja de Parrondo ha sido objeto de numerosas investigaciones desde su descubrimiento a finales del siglo pasado y ha mostrado tener diversas aplicaciones en muy distintos ámbitos. Esto es así por el hecho de que el fenómeno puede, en principio, aparecer en cualquier situación en la que se combinen dos dinámicas aleatorias a las que les podamos asociar una cierta función que sea no creciente según aquellas evolucionan.

Por ejemplo, en el campo de la teoría del control se ha tratado de relacionar la paradoja de Parrondo con el hecho de que la combinación de dos sistemas inestables puede hacer que se conviertan en estables. Una buena referencia en este sentido es [4].

Con respecto al campo de la biología, en [20] se menciona una situación real similar a alguna donde se puede dar la paradoja de Parrondo, ya que narra cómo a los pavos reales les interesa tener la cola cuanto más larga mejor porque así son más atractivos para las hembras, pese a que esto es una acción que los hace más vulnerables frente a los depredadores. También en [20] se trata de relacionar la paradoja de Parrondo con la formación de la proteína GCN4. Es interesante también el enfoque que se da en [42], donde se relaciona la paradoja de Parrondo con la genética de poblaciones, y en [9], donde se relaciona la paradoja con el comportamiento de los bacteriófagos.

La modelización de la paradoja de Parrondo sobre redes complejas hecha en la sección 5.1 también podría tener aplicaciones en ecología, pues podemos considerar una población de N individuos, cada uno ocupando un cierto espacio, donde su microhábitat social de

supervivencia viene definido por sus vecinos en la red. De esta manera, el juego A reflejaría la interacción competitiva entre un individuo y sus vecinos, mientras que el juego B reflejaría la dependencia que los individuos tienen en el microhábitat y las limitaciones que este impone sobre los individuos.

En el campo de la física, podríamos considerar la propia inspiración original de la paradoja de Parrondo, el movimiento browniano, como una de las posibles aplicaciones que tiene la paradoja, en este caso a una dinámica continua en el tiempo y en el espacio.

Actualmente se sigue estudiando la aportación de la paradoja de Parrondo en economía mediante la utilización de modelos simples sobre rendimientos de valores. En este contexto, la paradoja se aplica para demostrar que en determinados casos las inversiones individuales con rendimientos medios negativos a largo plazo se pueden combinar fácilmente en carteras diversificadas con rendimientos medios positivos a largo plazo, el llamado *volatility pumping*. Una buena fuente para profundizar en este tema es [44].

En [1], Abbott muestra algunas otras conexiones que la paradoja de Parrondo tiene con la física, la biología y la economía. La paradoja también ha inspirado trabajos en el estudio de la teoría de la fiabilidad, como podemos ver en [11], sincronización con ruido inducido, estudiada en [24], y en el control del caos, como en [45].

Por último, una de las aplicaciones que nos ha parecido más interesante es la que aparece en [8], donde se relaciona la paradoja con la pandemia de COVID-19 sufrida en el año 2020. En el artículo, se alterna entre dos estrategias sobre la población: la estrategia de imponer una cuarentena total y la estrategia de permitir el total movimiento entre individuos. Considerando una función de coste que tiene en cuenta el bienestar y la salud de la población así como el impacto económico del virus en la misma, ambas estrategias son negativas para la población, pues la primera daña gravemente la economía y la segunda eleva el número de infectados y fallecidos por el virus. Sin embargo, la alternancia entre ellas genera un beneficio para la población. Para modelizar la población se propone un modelo compartimental SIADÉ (cuyas siglas provienen de las iniciales de las palabras en inglés *susceptible, isolated, ailing, diagnosed* y *extinct*). Este modelo realiza una partición entre cinco compartimentos: susceptibles, aislados, enfermos no detectados, enfermos detectados y fallecidos. De esta manera, la primera estrategia propone un flujo del compartimento de los susceptibles a los aislados, mientras que la segunda propone un flujo en el sentido contrario. En el artículo se realizan diversas simulaciones que permiten observar la paradoja. También es interesante el enfoque que se da en [13], donde se muestra una potencial aplicación de la paradoja en el tratamiento de la diabetes tipo 2 mediante la alternancia de dos medicamentos, uno que baja los niveles de insulina del paciente y otro que los estimula.

Capítulo 6

Conclusiones

En este trabajo de fin de máster hemos centrado nuestra atención en el estudio detallado de la paradoja de Parrondo, un interesante fenómeno matemático cuyo análisis da lugar a un amplio número de cuestiones y líneas de investigación.

La memoria presenta un estudio detallado de diferentes variantes de la paradoja que ya aparecen en la literatura, como los juegos dependientes del capital, dependientes de la historia y los cooperativos. Además, hemos desarrollado programas numéricos en *Mathematica* y *Python* que resuelven los problemas matemáticos derivados del análisis de estos juegos.

De esta manera, hemos realizado un análisis del descubrimiento de la paradoja y probado su existencia en las tres situaciones posibles conocidas en la actualidad, realizando una profunda investigación de cada una de ellas y analizando las condiciones para que esta tenga lugar. Además, hemos obtenido nuevos resultados en el tema, como el estudio de la paradoja en función del parámetro de aleatorización y la existencia de la paradoja en la mezcla aleatoria de tres juegos. También hemos hecho referencia a algunas de las aplicaciones que la paradoja de Parrondo puede tener en diversos campos científicos.

Creemos con esto haber cumplido con creces los objetivos planteados en la introducción de este trabajo.

Como posibles líneas de investigación futuras planteamos la posible existencia de otros tipos de juegos donde se dé la paradoja, así como nuevas situaciones en las que aplicar la misma. En este sentido, una de las cuestiones más candentes en la actualidad en este tema es la relación de la paradoja de Parrondo con la teoría de juegos cuánticos. Una buena referencia para el lector interesado en este tema es [28]. También es interesante la idea que se plasma en [50] y [51], donde se lleva la paradoja de Parrondo en juegos cooperativos a una malla bidimensional, de manera que cada jugador tiene 8 vecinos posibles. La investigación futura de este tema y su relación con los autómatas celulares podría dar lugar a interesantes resultados.

Otro enfoque interesante es el que se da en [21] y [22], donde se estudia la existencia de la paradoja de Parrondo en el caso en el que la pérdida de la probabilidad ϵ no es constante sino que está definida como una función periódica en el tiempo. Esta situación podría ser de utilidad en el área de la dinámica de poblaciones. Planteamos que pueda haber algún organismo que base su supervivencia en la alternancia en el tiempo entre dos dinámicas en principio perjudiciales, como por ejemplo ser nómada y asentarse en un sitio fijo, de manera que dicha alternancia permita la supervivencia de la especie. El hecho de

que en muchas poblaciones de seres vivos se den ciclos estacionarios podría permitir la introducción de dicho factor ϵ periódico.

También resulta atractiva la idea de relacionar el estudio de la paradoja de Parrondo visto en el capítulo 5 con la paradoja de Braess [26], según la cual añadir una o más carreteras a una red de carreteras puede acabar dificultando el flujo de tráfico general a través de ella, debido a que cada conductor tomará el camino que le resulte aparentemente más favorable, aumentando así los tiempos de espera y empeorando el rendimiento de la red.

Además, sería importante aplicar en un entorno con una gran capacidad de computación los códigos creados, y así poder extender el estudio de este trabajo para ampliar los resultados presentes y alcanzar algunos nuevos. Como desarrollo futuro, sería interesante la continuación de las novedades introducidas en este trabajo. Por ejemplo, sería de utilidad aplicar las investigaciones realizadas en la sección 5.1 a otro tipo de redes complejas y ver si se alcanzan conclusiones similares a las alcanzadas allí, así como estudiar la existencia de la paradoja de Parrondo en la mezcla aleatoria de cuatro o más juegos.

Capítulo 7

Anexos

7.1. Anexo 1: Códigos de *Python*

7.1.1. Potencial asimétrico de dientes de sierra al que se ven sometidas las partículas brownianas

```
import numpy as np
import matplotlib.pyplot as plt

def potential(a, V0, x, L):
    x_modulo = np.mod(x, L)
    return np.where(x_modulo <= a, V0 * x_modulo / a, V0 - V0 * (x_modulo - a) / (1-a))

L = 1
a = 1/3
V0 = 5
x = np.linspace(0, 3, 1000)
V = potential(a, V0, x, L)

plt.figure(figsize=(10, 5))
plt.plot(x, V)
plt.xlabel('$x$')
plt.ylabel('$V(x)$')
plt.show()
```

7.1.2. Evolución del capital para el juego original *A*

```
# Importamos la biblioteca random para hacer los cálculos aleatorios
import random
# Importamos la biblioteca matplotlib para los gráficos
import matplotlib.pyplot as plt

# Definimos una función que ahora solo juego el juego A, pero posteriormente usaremos para
# elegir aleatoriamente el juego A o el B, cada uno con una probabilidad del 50 %
def jugar(juego, capital, p):
    if juego == 'A':
        # Si el juego es A, el jugador tiene una probabilidad p de ganar
        # random.random() da un número real aleatorio entre 0 y 1
        if random.random() < p:
            # Si gana, aumentamos el capital en una unidad
            capital += 1
        else:
            # Si pierde, disminuimos el capital en una unidad
            capital -= 1
    return capital

# Inicializamos una lista para almacenar el capital después de cada iteración
```

```

capitalesA = []

# Configuramos los parámetros del juego
epsilon = 0.005
p = 0.5 - epsilon

# Configuramos el capital inicial
capital = 0

# Ejecutamos 100 iteraciones del juego
for _ in range(100):
    juego = "A"
    capital = jugar(juego, capital, p)
    capitalesA.append(capital)

# Visualizamos la evolución del capital
plt.plot(capitalesA)
plt.xlabel('Iteración')
plt.ylabel('Capital')
plt.title('Evolución del capital a lo largo de 100 iteraciones')
plt.show()

```

7.1.3. Evolución del capital para el juego original B

```

import random
import matplotlib.pyplot as plt

def jugar(juego, capital, multiplo, pm, pb):
    if juego == 'B':
        # Si el juego es B, el jugador mira si su capital es múltiplo de M
        if capital % multiplo == 0:
            # Si el capital es múltiplo de M, tiene una probabilidad pm de ganar
            if random.random() < pm:
                # Aumentamos su capital en una unidad
                capital += 1
            else:
                capital -= 1
        else:
            # Si el capital no es múltiplo de la variable, tiene una probabilidad pb de ganar
            if random.random() < pb:
                capital += 1
            else:
                capital -= 1

    return capital

capitalesB = []
epsilon = 0.005
pm = 0.1 - epsilon
pb = 0.75 - epsilon
multiplo = 3
capital = 0

for _ in range(100):
    juego = "B"
    capital = jugar(juego, capital, multiplo, pm, pb)
    capitalesB.append(capital)

plt.plot(capitalesB)
plt.xlabel('Iteración')
plt.ylabel('Capital')
plt.title('Evolución del capital a lo largo de 100 iteraciones')
plt.show()

```

7.1.4. Evolución del capital para el caso original $A + B$

```

import random
import matplotlib.pyplot as plt

```

```

def elegir_juego():
    juegos = ['A', 'B']
    # Elige aleatoriamente el juego
    juego_elegido = random.choice(juegos)
    # Devuelve el juego elegido
    return juego_elegido

# Definimos una función que juega al juego elegido de manera aleatoria anterior.
def jugar(juego, capital, multiplo, p, pm, pb):
    if juego == 'A':
        if random.random() < p:
            capital += 1
        else:
            capital -= 1
    elif juego == 'B':
        # Si el juego es B, el jugador mira si su capital es múltiplo de M
        if capital % multiplo == 0:
            # Si el capital es múltiplo de M, tiene una probabilidad pm de ganar
            if random.random() < pm:
                # Aumentamos su capital en una unidad
                capital += 1
            else:
                capital -= 1
        else:
            # Si el capital no es múltiplo de la variable, tiene una probabilidad pb de ganar
            if random.random() < pb:
                capital += 1
            else:
                capital -= 1
    return capital

# Inicializamos una lista que almacena el capital después de cada iteración
capitales = []

# Configuramos los parámetros del juego
epsilon = 0.005
p = 0.5 - epsilon
p1 = 0.1 - epsilon
p2 = 0.75 - epsilon
multiplo = 3

# Configuramos el capital inicial
capital = 0

# Ejecutamos 100 iteraciones del juego
for _ in range(100):
    juego = elegir_juego()
    capital = jugar(juego, capital, multiplo, p, p1, p2)
    capitales.append(capital)

# Visualizamos la evolución del capital
plt.plot(capitales)
plt.xlabel('Iteración')
plt.ylabel('Capital')
plt.title('Evolución del capital a lo largo de 100 iteraciones')
plt.show()

```

7.1.5. Capital esperado en el juego original A

```

A = np.array([
    [0, 0.5-epsilon, 0.5+epsilon],
    [0.5+epsilon, 0, 0.5-epsilon],
    [0.5-epsilon, 0.5+epsilon, 0]
])

distribucion_inicial = np.array([1/3, 1/3, 1/3])

def proxima_distribucion(distribucion_actual, P):
    return distribucion_actual.dot(P)

```

```

def prob_variable_1(distribucion_actual):
    return (0.5-epsilon) * distribucion_actual[0] + \
        (0.5-epsilon) * distribucion_actual[1] + (0.5-epsilon) * distribucion_actual[2]

num_iteraciones = 100

distribucion_actual = distribucion_inicial

esperanza_variable = []
capital = [0]

for i in range(num_iteraciones):
    siguiente_distribucion = proxima_distribucion(distribucion_actual, A)

    # Calcular la probabilidad de que la variable aleatoria valga 1
    p_1 = prob_variable_1(siguiente_distribucion)

    # Calcular la esperanza de la variable aleatoria
    esperanza = 1 * p_1 + (-1) * (1 - p_1)

    # Almacenar la esperanza en la lista
    esperanza_variable.append(esperanza)

    # Actualizar el capital sumando el valor de la esperanza actual
    capital_nuevo = capital[-1] + esperanza
    capital.append(capital_nuevo)

    # Actualizar la distribución actual
    distribucion_actual = siguiente_distribucion

plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 2)
plt.plot(capital)
plt.xlabel('Iteraciones')
plt.ylabel('Capital')
plt.title('Evolución del Capital en el juego A')
plt.grid(True)

plt.tight_layout()
plt.show()

```

7.1.6. Capital esperado en el juego original B

```

import numpy as np
import matplotlib.pyplot as plt

epsilon = 0.005

# Definimos la matriz B
B = np.array([
    [0, 0.1-epsilon, 0.9+epsilon],
    [0.25+epsilon, 0, 0.75-epsilon],
    [0.75-epsilon, 0.25+epsilon, 0]
])

distribucion_inicial = np.array([1/3, 1/3, 1/3])

# Función para calcular la distribución en el siguiente paso
def proxima_distribucion(distribucion_actual, P):
    return distribucion_actual.dot(P)

# Función para calcular la probabilidad de que la variable aleatoria valga 1
def prob_variable_1(distribucion_actual):
    return (0.1-epsilon) * distribucion_actual[0] + \
        (0.75-epsilon) * distribucion_actual[1] + (0.75-epsilon) * distribucion_actual[2]

num_iteraciones = 100

```

```

# Inicializamos la distribución actual
distribucion_actual = distribucion_inicial

esperanza_variable = []
capital = [0]

# Bucle para iterar la evolución
for i in range(num_iteraciones):
    # Calculamos la nueva distribución
    siguiente_distribucion = proxima_distribucion(distribucion_actual, B)

    # Calculamos la probabilidad de que la variable aleatoria valga 1
    p_1 = prob_variable_1(siguiente_distribucion)

    # Calculamos la esperanza de la variable aleatoria
    esperanza = 1 * p_1 + (-1) * (1 - p_1)

    # Almacenamos la esperanza en la lista
    esperanza_variable.append(esperanza)

    # Actualizamos el capital sumando el valor de la esperanza actual
    capital_nuevo = capital[-1] + esperanza
    capital.append(capital_nuevo)

    # Actualizar la distribución actual
    distribucion_actual = siguiente_distribucion

# Graficar el capital a lo largo de las iteraciones
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 2)
plt.plot(capital)
plt.xlabel('Iteraciones')
plt.ylabel('Capital')
plt.title('Evolución del Capital en el juego B')
plt.grid(True)

plt.tight_layout()
plt.show()

```

7.1.7. Capital esperado en el juego original $A + B$

```

A = np.array([
    [0, 0.5-epsilon, 0.5+epsilon],
    [0.5+epsilon, 0, 0.5-epsilon],
    [0.5-epsilon, 0.5+epsilon, 0]
])

B = np.array([
    [0, 0.1-epsilon, 0.9+epsilon],
    [0.25+epsilon, 0, 0.75-epsilon],
    [0.75-epsilon, 0.25+epsilon, 0]
])

P = (1/2)*(A+B)

distribucion_inicial = np.array([1/3, 1/3, 1/3])

def proxima_distribucion(distribucion_actual, P):
    return distribucion_actual.dot(P)

def prob_variable_1(distribucion_actual):
    return 0.5*(0.5-epsilon) + 0.5*((0.1-epsilon) * distribucion_actual[0]) + \
        0.5*((0.75-epsilon) * distribucion_actual[1]) + \
        0.5*((0.75-epsilon) * distribucion_actual[2])

num_iteraciones = 100

distribucion_actual = distribucion_inicial

```

```

esperanza_variable = []
capital = [0]

for i in range(num_iteraciones):
    siguiente_distribucion = proxima_distribucion(distribucion_actual, P)

    # Calcular la probabilidad de que la variable aleatoria valga 1
    p_1 = prob_variabile_1(siguiente_distribucion)

    # Calcular la esperanza de la variable aleatoria
    esperanza = 1 * p_1 + (-1) * (1 - p_1)

    # Almacenar la esperanza en la lista
    esperanza_variable.append(esperanza)

    # Actualizar el capital sumando el valor de la esperanza actual
    capital_nuevo = capital[-1] + esperanza
    capital.append(capital_nuevo)

    # Actualizar la distribución actual
    distribucion_actual = siguiente_distribucion

plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 2)
plt.plot(capital)
plt.xlabel('Iteraciones')
plt.ylabel('Capital')
plt.title('Evolución del Capital en el juego A + B')
plt.grid(True)

plt.tight_layout()
plt.show()

```

7.1.8. Evolucion promedio del capital en funcion del capital inicial

```

import random
import matplotlib.pyplot as plt
import numpy as np

def elegir_juego():
    juegos = ['A', 'B']
    juego_elegido = random.choice(juegos)
    return juego_elegido

def jugar(juego, capital, multiplo, p, p1, p2):
    if juego == 'A':
        if random.random() < p:
            capital += 1
        else:
            capital -= 1
    elif juego == 'B':
        if capital % multiplo == 0:
            if random.random() < p1:
                capital += 1
            else:
                capital -= 1
        else:
            if random.random() < p2:
                capital += 1
            else:
                capital -= 1
    return capital

epsilon = 0.005
p = 0.5 - epsilon
p1 = 0.1 - epsilon
p2 = 0.75 - epsilon

```

```

multiplo = 3

num_simulaciones = 100
num_iteraciones = 100

# Inicializar un diccionario para almacenar las evoluciones del capital para cada capital inicial
resultados = {0: [], 1: [], 2: [], 3: []}

for capital_inicial in resultados.keys():
    for _ in range(num_simulaciones):
        # Inicializamos lista para almacenar el capital después de cada iteración
        capitales = [capital_inicial]
        # Configuramos el capital inicial
        capital = capital_inicial
        for _ in range(num_iteraciones):
            juego = elegir_juego()
            capital = jugar(juego, capital, multiplo, p, p1, p2)
            capitales.append(capital)
        # Almacenamos la evolución de esta simulación
        resultados[capital_inicial].append(capitales)

# Calcular el promedio del capital en cada iteración para cada capital inicial
promedios = {k: np.mean(v, axis=0) for k, v in resultados.items()}

# Visualizamos la evolución promedio del capital para cada capital inicial
plt.figure(figsize=(12, 8))

for capital_inicial, promedio_capitales in promedios.items():
    plt.plot(promedio_capitales, label=f'Capital inicial = {capital_inicial}')

plt.xlabel('Iteración')
plt.ylabel('Capital promedio')
plt.title('Evolución promedio del capital')
plt.legend()
plt.savefig('evolucion_capital.jpg', format='jpg')
plt.show()

```

7.1.9. Evolucion promedio del capital para diferentes valores (a, b)

```

import random
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

def jugar(juego, capital, multiplo, p, p1, p2):
    if juego == 'A':
        if random.random() < p:
            capital += 1
        else:
            capital -= 1
    elif juego == 'B':
        if capital % multiplo == 0:
            if random.random() < p1:
                capital += 1
            else:
                capital -= 1
        else:
            if random.random() < p2:
                capital += 1
            else:
                capital -= 1
    return capital

epsilon = 0.005
p = 0.5 - epsilon
p1 = 0.1 - epsilon
p2 = 0.75 - epsilon

```

```

multiplo = 3

valores_a = range(1, 11) # Por ejemplo, de 1 a 10
valores_b = range(1, 11)

# Inicializamos una matriz para almacenar el capital final promedio para cada par (a, b)
capital_promedio = np.zeros((len(valores_a), len(valores_b)))

num_simulaciones = 100

# Ejecutamos simulaciones para cada combinación de (a, b) y calculamos el promedio
for i, a in enumerate(valores_a):
    for j, b in enumerate(valores_b):
        capital_suma = 0
        for _ in range(num_simulaciones):
            capital = 0
            # Jugar A a veces
            for _ in range(a):
                capital = jugar('A', capital, multiplo, p, p1, p2)
            # Jugar B b veces
            for _ in range(b):
                capital = jugar('B', capital, multiplo, p, p1, p2)
            # Aquí se almacena el resultado a lo largo de todas las iteraciones
            capital_suma += capital
        # Calculamos el promedio del capital final
        capital_promedio[i, j] = capital_suma / num_simulaciones

# Creamos el gráfico 3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
A, B = np.meshgrid(valores_a, valores_b)

# Invertimos el orden en el eje b
ax.plot_surface(A, B, capital_promedio.T, cmap='viridis')
ax.invert_yaxis()

ax.set_xlabel('a')
ax.set_ylabel('b')
ax.set_zlabel('Capital Promedio')
ax.set_title('Evolución del capital promedio para diferentes secuencias (a, b)')
plt.savefig('evolucion_capital_promedio_3d.jpg', format='jpg')

plt.show()

```

7.1.10. Capital esperado en el juego *ABB*

```

import numpy as np
import matplotlib.pyplot as plt

epsilon = 0.005

PA = np.array([
    [0, 0.5-epsilon, 0.5+epsilon],
    [0.5+epsilon, 0, 0.5-epsilon],
    [0.5-epsilon, 0.5+epsilon, 0]
])

PB = np.array([
    [0, 0.1-epsilon, 0.9+epsilon],
    [0.25+epsilon, 0, 0.75-epsilon],
    [0.75-epsilon, 0.25+epsilon, 0]
])

def elegir_matriz(i):
    if i%3==0:
        return PA
    if i%3==1:
        return PB
    if i%3==2:

```

```

    return PB

distribucion_inicial = np.array([1/3, 1/3, 1/3])

# Función para calcular la distribución en el siguiente paso
def distribucion_siguiete(distribucion_actual, P):
    return distribucion_actual.dot(P)

# Función para calcular la probabilidad de que la variable aleatoria valga 1
def prob_variable_1(distribucion_actual, i):
    if i%3==0:
        return (0.5-epsilon)
    if i%3==1:
        return ((0.1-epsilon) * distribucion_actual[0]) + \
            ((0.75-epsilon) * distribucion_actual[1]) + ((0.75-epsilon) * \
            distribucion_actual[2])
    if i%3==2:
        return ((0.1-epsilon) * distribucion_actual[0]) + \
            ((0.75-epsilon) * distribucion_actual[1]) + ((0.75-epsilon) * \
            distribucion_actual[2])

# Parámetros para la simulación
num_iteraciones = 100
tolerancia = 1e-6

# Inicializar la distribución actual
distribucion_actual = distribucion_inicial

esperanza_variable = []
capital = [0] # Lista para almacenar los valores de Capital, iniciando en 0

for i in range(num_iteraciones):

    P = elegir_matriz(i)

    distribucion_sig = distribucion_siguiete(distribucion_actual, P)

    # Calcular la probabilidad de que la variable aleatoria valga 1
    p_1 = prob_variable_1(distribucion_actual, i)

    # Calcular la esperanza de la variable aleatoria
    esperanza = 1 * p_1 + (-1) * (1 - p_1)

    # Almacenar la esperanza en la lista
    esperanza_variable.append(esperanza)

    # Actualizar el capital sumando el valor de la esperanza actual
    capital_nuevo = capital[-1] + esperanza
    capital.append(capital_nuevo)

    # Actualizar la distribución actual
    distribucion_actual = distribucion_sig
    print(distribucion_actual)

plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 2)
plt.plot(capital)
plt.xlabel('Iteraciones')
plt.ylabel('Capital')
plt.title('Evolución del Capital en el juego ABBABB...')
plt.grid(True)

plt.tight_layout()
plt.show()

```

7.1.11. Evolución del capital del juego B dependiente de la historia

```
import random
import matplotlib.pyplot as plt

# Definimos una función que aleatoriamente asigna al jugador su historial inicial,
# 0 para derrota, 1 para victoria

def asignar_historia():
    valores = []
    for _ in range(2):
        valor = random.randint(0, 1)
        valores.append(valor)
    return valores

def jugar(juego, capital, historia, p, p1, p2, p3, p4):
    if juego == 'A':
        if random.random() < p:
            historia.append(1)
            capital += 1
        else:
            historia.append(0)
            capital -= 1
        return capital
    elif juego == 'B':
        if historia[-1] == 0 and historia[-2] == 0:
            probabilidad = p1
        elif historia[-1] == 0 and historia[-2] == 1:
            probabilidad = p2
        elif historia[-1] == 1 and historia[-2] == 0:
            probabilidad = p3
        else:
            probabilidad = p4

        if random.random() < probabilidad:
            capital += 1
        else:
            capital -= 1

    return capital

capital = 0
historia = asignar_historia()
evolucion_capital = [capital]
iteraciones = list(range(1, 21))

e = 0.005
p = 1/2 - e
p1 = 9/10 - e
p2 = 1/4 - e
p3 = 1/4 - e
p4 = 7/10 - e

for _ in range(100):
    juego = "B"
    capital = jugar(juego, capital, historia, p, p1, p2, p3, p4)
    evolucion_capital.append(capital)

evolucion_capital.pop()

plt.plot(evolucion_capital)
plt.xlabel('Iteración')
plt.ylabel('Capital')
plt.title('Evolución del capital en 100 iteraciones')
plt.grid(True)
plt.show()
```

7.1.12. Evolución del capital para el caso dependiente de la historia $A + B$

```
import random
import matplotlib.pyplot as plt

def asignar_historia():
    valores = []
    for _ in range(2):
        valor = random.randint(0, 1)
        valores.append(valor)
    return valores

def elegir_juego():
    juegos = ['A', 'B']
    return random.choice(juegos)

# Definimos una función que a partir del juego elegido, la historia del jugador
# y las probabilidades del juego B hace que el jugador juegue uno de los dos juegos
def jugar(juego, capital, historia, p, p1, p2, p3, p4):
    if juego == 'A':
        if random.random() < p:
            historia.append(1)
            capital += 1
        else:
            historia.append(0)
            capital -= 1
        return capital
    elif juego == 'B':
        if historia[-1] == 0 and historia[-2] == 0:
            probabilidad = p1
        elif historia[-1] == 0 and historia[-2] == 1:
            probabilidad = p2
        elif historia[-1] == 1 and historia[-2] == 0:
            probabilidad = p3
        else:
            probabilidad = p4

        if random.random() < probabilidad:
            capital += 1
        else:
            capital -= 1

    return capital

capital = 0
historia = asignar_historia()
evolucion_capital = [capital]
iteraciones = list(range(1, 21))

e = 0.005
p = 1/2 - e
p1 = 9/10 - e
p2 = 1/4 - e
p3 = 1/4 - e
p4 = 7/10 - e

for _ in range(100):
    juego = elegir_juego()
    capital = jugar(juego, capital, historia, p, p1, p2, p3, p4)
    evolucion_capital.append(capital)

evolucion_capital.pop()

plt.plot(evolucion_capital)
plt.xlabel('Iteración')
plt.ylabel('Capital')
plt.title('Evolución del capital en 100 iteraciones')
plt.grid(True)
```

```
plt.show()
```

7.1.13. Evolución del capital para el juego B cooperativo

```
import random
import matplotlib.pyplot as plt

# Definimos una función que a partir del número de jugadores asigna un 0 o un 1
# a cada uno, dependiendo de si en su anterior juego
# resultaron ganadores o perdedores, de manera aleatoria
def asignar_valores(n_jugadores):
    valores = []
    for _ in range(n_jugadores):
        valor = random.randint(0, 1)
        valores.append(valor)
    return valores

# Definimos la función que, a partir de la cantidad de jugadores perdedores o ganadores
# existentes al empezar, calcula el capital inicial con el que se empieza a jugar,
# sumando 1 por cada ganador y restándolo por cada perdedor
def calcular_capital_inicial(valores_jugadores, capital):
    for valor in valores_jugadores:
        if valor == 1:
            capital += 1
        else:
            capital -= 1
    return capital

# Definimos una función que elige un jugador aleatoriamente de entre todos
def elegir_jugador(n_jugadores):
    return random.randint(0, n_jugadores - 1)
# Hemos generado un número aleatorio entre 0 y n_jugadores - 1

# Definimos la función que va actualizando el capital en las demás iteraciones
def calcular_capital(valores_jugadores, capital, jugador_elegido):
    if valores_jugadores[jugador_elegido] == 1:
        capital += 1
    else:
        capital -= 1
    return capital

# Definimos una función que a partir del jugador elegido, los valores de los jugadores
# y las probabilidades del juego B hace que el jugador juegue el juego B
def jugar(jugador_elegido, juego, valores_jugadores, p, p0, p1, p2, p3):
    if juego == 'B':
        # Si el juego es B, se determina el nuevo valor del jugador basado en
        # los estados de sus vecinos
        izquierda = valores_jugadores[(jugador_elegido - 1) % len(valores_jugadores)]
        # El módulo se utiliza para no salirnos de los índices, e ir al último jugador
        derecha = valores_jugadores[(jugador_elegido + 1) % len(valores_jugadores)]

        if izquierda == 0 and derecha == 0:
            probabilidad = p0
        elif izquierda == 0 and derecha == 1:
            probabilidad = p1
        elif izquierda == 1 and derecha == 0:
            probabilidad = p2
        else:
            probabilidad = p3

        # Elige entre 0 o 1 en función de los pesos dados por la probabilidad
        # que se esté empleando en B
        nuevo_valor = random.choices([0, 1], weights=[1-probabilidad, probabilidad])[0]
        return nuevo_valor

capital_actual = 0 # Capital inicial
n_jugadores = 100
valores_jugadores = asignar_valores(n_jugadores)
```

```

capital_actual = calcular_capital_inicial(valores_jugadores, capital_actual)
# Usamos la función para calcular el capital inicial

evolucion_capital = [capital_actual]
iteraciones = list(range(1, 21)) # Hacemos 20 iteraciones

e = 0.005
p0 = 1/4 - e
p1 = 6/10 - e
p2 = 6/10 - e
p3 = 5/8 - e

for _ in range(20):
    jugador_elegido = elegir_jugador(n_jugadores)
    juego_elegido = "B"
    valores_jugadores[jugador_elegido] = jugar(jugador_elegido, juego_elegido,
                                                valores_jugadores, p, p0, p1, p2, p3)
    capital_actual = calcular_capital(valores_jugadores, capital_actual, jugador_elegido)
    evolucion_capital.append(capital_actual)

# Eliminar el último elemento de evolucion_capital, para que no de fallo
evolucion_capital.pop()

plt.plot(iteraciones, evolucion_capital)
plt.xlabel('Iteración')
plt.ylabel('Capital')
plt.title('Evolución del capital en 100 iteraciones')
plt.grid(True)
plt.show()

```

7.1.14. Evolución del capital para el juego $A + B$ cooperativo

```

import random
import matplotlib.pyplot as plt

def asignar_valores(n_jugadores):
    valores = []
    for _ in range(n_jugadores):
        valor = random.randint(0, 1)
        valores.append(valor)
    return valores

def calcular_capital_inicial(valores_jugadores, capital):
    for valor in valores_jugadores:
        if valor == 1:
            capital += 1
        else:
            capital -= 1
    return capital

def elegir_jugador(n_jugadores):
    return random.randint(0, n_jugadores - 1)

def calcular_capital(valores_jugadores, capital, jugador_elegido):
    if valores_jugadores[jugador_elegido] == 1:
        capital += 1
    else:
        capital -= 1
    return capital

# Definimos una función que elige un juego aleatoriamente
def elegir_juego():
    juegos = ['A', 'B']
    return random.choice(juegos)

# Definimos una función que a partir del jugador elegido, el juego elegido, los

```

```

# valores de los jugadores y las probabilidades del juego B hace que el jugador juegue
# uno de los dos juegos
def jugar(jugador_elegido, juego, valores_jugadores, p, p0, p1, p2, p3):
    if juego == 'A':
        if random.random() < p:
            jugador_nuevo_valor = 1
        else:
            jugador_nuevo_valor = 0
        return jugador_nuevo_valor
    elif juego == 'B':
        izquierda = valores_jugadores[(jugador_elegido - 1) % len(valores_jugadores)]
        derecha = valores_jugadores[(jugador_elegido + 1) % len(valores_jugadores)]

        if izquierda == 0 and derecha == 0:
            probabilidad = p0
        elif izquierda == 0 and derecha == 1:
            probabilidad = p1
        elif izquierda == 1 and derecha == 0:
            probabilidad = p2
        else:
            probabilidad = p3

        nuevo_valor = random.choices([0, 1], weights=[1-probabilidad, probabilidad])[0]
        return nuevo_valor

capital_actual = 0
n_jugadores = 100
valores_jugadores = asignar_valores(n_jugadores)

capital_actual = calcular_capital_inicial(valores_jugadores, capital_actual)

evolucion_capital = [capital_actual]
iteraciones = list(range(1, 21))

e = 0.005
p = 1/2 - e
p0 = 1/4 - e
p1 = 6/10 - e
p2 = 6/10 - e
p3 = 5/8 - e

for _ in range(20):
    jugador_elegido = elegir_jugador(n_jugadores)
    juego_elegido = elegir_juego()
    valores_jugadores[jugador_elegido] = jugar(jugador_elegido, juego_elegido, valores_jugadores, p, p0, p1, p2, p3)
    capital_actual = calcular_capital(valores_jugadores, capital_actual, jugador_elegido)
    evolucion_capital.append(capital_actual)

evolucion_capital.pop()

plt.plot(iteraciones, evolucion_capital)
plt.xlabel('Iteración')
plt.ylabel('Capital')
plt.title('Evolución del capital en 100 iteraciones')
plt.grid(True)
plt.show()

```

7.1.15. Evolución del capital para una red completa con el patrón 1 de comportamiento

```

# Importación de bibliotecas
import matplotlib.pyplot as plt
import random
import numpy as np

# Función para obtener vecinos en un grafo totalmente conectado
def conseguir_vecinos(indice_jugador, numero_jugadores):

```

```

    return [j for j in range(numero_jugadores) if j != indice_jugador]

# Parámetros del problema
numero_jugadores = 500
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 10
capital_inicial = 10
e = 0.005
p = 0.5 # Probabilidad de jugar al juego A
p1 = 0.1 - e # Probabilidad de ganar el juego B si el capital es múltiplo de 3
p2 = 0.75 - e # Probabilidad de ganar el juego B si el capital no es múltiplo de 3

# Inicialización del vector para la evolución media entre todas las simulaciones del capital total
# en cada iteración
evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

# Realizar las simulaciones
for sim in range(numero_simulaciones):
    # Inicialización de los capitales de los jugadores para esta simulación
    capitales = [capital_inicial] * numero_jugadores

    # Simulación de las iteraciones para esta simulación
    for iteracion in range(numero_iteraciones_por_simulacion):
        # Elegir aleatoriamente un jugador
        i = random.randint(0, numero_jugadores - 1)

        # Decisión de jugar A o B
        if random.random() < p:
            # Juego A: Aplicamos el patrón de competición
            vecinos = conseguir_vecinos(i, numero_jugadores)
            j = random.choice(vecinos)

            # Ambos jugadores tienen probabilidad 0.5 de ganar
            if random.random() < 0.5:
                # i gana, j paga una unidad a i
                capitales[i] += 1
                capitales[j] -= 1
            else:
                # j gana, i paga una unidad a j
                capitales[i] -= 1
                capitales[j] += 1
        else:
            # Juego B
            if capitales[i] % 3 == 0:
                # El capital de i es múltiplo de 3
                if random.random() < p1:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                # El capital de i no es múltiplo de 3
                if random.random() < p2:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1

        # Calcular el capital total al final de esta iteración
        capital_total = sum(capitales)

        # Agregar el capital total al vector de evolución media
        evolucion_media[iteracion] += capital_total / numero_simulaciones

# Imprimir la evolución media del capital total
print("Evolución media del capital total:")
print(evolucion_media)

# Número de iteraciones
iterations = list(range(1, len(evolucion_media) + 1))
# Creamos el gráfico de puntos
plt.scatter(iterations, evolucion_media, color='blue')

```

```
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()
```

7.1.16. Evolución del capital para una red completa con el patrón 2 de comportamiento

```
import matplotlib.pyplot as plt
import random
import numpy as np

def conseguir_vecinos(indice_jugador, numero_jugadores):
    return [j for j in range(numero_jugadores) if j != indice_jugador]

numero_jugadores = 500
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 10
capital_inicial = 10
e = 0.005
p = 0.5
p1 = 0.1 - e
p2 = 0.75 - e

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if random.random() < p:
            # Juego A: transferencia de una unidad de capital de i a j
            neighbors = conseguir_vecinos(i, numero_jugadores)
            j = random.choice(neighbors)
            capitales[i] -= 1
            capitales[j] += 1
        else:
            if capitales[i] % 3 == 0:
                if random.random() < p1:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if random.random() < p2:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1

    capital_total = sum(capitales)

    evolucion_media[iteracion] += capital_total / numero_simulaciones

print("Evolución media del capital total:")
print(evolucion_media)

iterations = list(range(1, len(evolucion_media) + 1))
plt.scatter(iterations, evolucion_media, color='blue')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()
```

7.1.17. Evolución del capital para una red completa con el patrón 3 de comportamiento

```
import matplotlib.pyplot as plt
import random
import numpy as np

def conseguir_vecinos(indice_jugador, numero_jugadores):
    return [j for j in range(numero_jugadores) if j != indice_jugador]

numero_jugadores = 500
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 10
capital_inicial = 10
e = 0.005
p = 0.5
p1 = 0.1 - e
p2 = 0.75 - e

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if random.random() < p:
            vecinos = conseguir_vecinos(i, numero_jugadores)
            j = random.choice(vecinos)

            if capitales[i] >= capitales[j]:
                # i paga una unidad a j
                capitales[i] -= 1
                capitales[j] += 1
            else:
                capitales[i] += 1
                capitales[j] -= 1
        else:
            if capitales[i] % 3 == 0:
                if random.random() < p1:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if random.random() < p2:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1

    capital_total = sum(capitales)

    evolucion_media[iteracion] += capital_total / numero_simulaciones

print("Evolución media del capital total:")
print(evolucion_media)

iterations = list(range(1, len(evolucion_media) + 1))
plt.scatter(iterations, evolucion_media, color='blue')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()
```

7.1.18. Evolución del capital para una red completa con el patrón 4 de comportamiento

```
import matplotlib.pyplot as plt
import random
import numpy as np

def conseguir_vecinos(indice_jugador, numero_jugadores):
    return [j for j in range(numero_jugadores) if j != indice_jugador]

numero_jugadores = 500
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 10
capital_inicial = 10
e = 0.005
p = 0.5
p1 = 0.1 - e
p2 = 0.75 - e

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if random.random() < p:
            vecinos = conseguir_vecinos(i, num_players)
            j = random.choice(vecinos)

            if capitales[i] >= capital_inicial:
                # i coopera: i paga una unidad a j
                capitales[i] -= 1
                capitales[j] += 1
            else:
                # i compete: se selecciona aleatoriamente quién gana
                if random.random() < 0.5:
                    # i gana, j paga una unidad a i
                    capitales[i] += 1
                    capitales[j] -= 1
                else:
                    # j gana, i paga una unidad a j
                    capitales[i] -= 1
                    capitales[j] += 1
            else:
                if capitales[i] % 3 == 0:
                    if random.random() < p1:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1
                else:
                    if random.random() < p2:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1

    capital_total = sum(capitales)

    evolucion_media[iteracion] += capital_total / numero_simulaciones

print("Evolución media del capital total:")
print(evolucion_media)

iterations = list(range(1, len(evolucion_media) + 1))
plt.scatter(iterations, evolucion_media, color='blue')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
```

```
plt.grid(True)
plt.show()
```

7.1.19. Evolución del capital para una red de mundo pequeño con el patrón 1 de comportamiento

```
# Importación de bibliotecas
import matplotlib.pyplot as plt
import random
import networkx as nx
import numpy as np

numero_jugadores = 500
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 10
capital_inicial = 10
e = 0.005
p = 0.5
p1 = 0.1 - e
p2 = 0.75 - e

# Creamos el grafo de mundo pequeño usando el modelo de Watts-Strogatz
k = 10 # Cada nodo se conecta a k vecinos
p_ws = 0.1 # Probabilidad de reconexión
G = nx.watts_strogatz_graph(numero_jugadores, k, p_ws)

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if random.random() < p:
            # Obtenemos los vecinos
            vecinos = list(G.neighbors(i))
            j = random.choice(vecinos)

            if random.random() < 0.5:
                capitales[i] += 1
                capitales[j] -= 1
            else:
                capitales[i] -= 1
                capitales[j] += 1
        else:
            if capitales[i] % 3 == 0:
                if random.random() < p1:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if random.random() < p2:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1

    capital_total = sum(capitales)

    evolucion_media[iteracion] += capital_total / numero_simulaciones

print("Evolución media del capital total:")
print(evolucion_media)

iterations = list(range(1, len(evolucion_media) + 1))
plt.scatter(iterations, evolucion_media, color='blue')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
```

```
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()
```

7.1.20. Evolución del capital para una red de mundo pequeño con el patrón 2 de comportamiento

```
import matplotlib.pyplot as plt
import random
import networkx as nx
import numpy as np

numero_jugadores = 500
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 10
capital_inicial = 10
e = 0.005
p = 0.5
p1 = 0.1 - e
p2 = 0.75 - e

# Creamos el grafo de mundo pequeño usando el modelo de Watts-Strogatz
k = 10 # Cada nodo se conecta a k vecinos
p_ws = 0.1 # Probabilidad de reconexión
G = nx.watts_strogatz_graph(numero_jugadores, k, p_ws)

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if random.random() < p:
            vecinos = list(G.neighbors(i))
            j = random.choice(vecinos)

            capitales[i] -= 1
            capitales[j] += 1
        else:
            if capitales[i] % 3 == 0:
                if random.random() < p1:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if random.random() < p2:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1

        capital_total = sum(capitales)

        evolucion_media[iteracion] += capital_total / numero_simulaciones

print("Evolución media del capital total:")
print(evolucion_media)

iterations = list(range(1, len(evolucion_media) + 1))
plt.scatter(iterations, evolucion_media, color='blue')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()
```

7.1.21. Evolución del capital para una red de mundo pequeño con el patrón 3 de comportamiento

```
import matplotlib.pyplot as plt
import random
import networkx as nx
import numpy as np

numero_jugadores = 500
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 10
capital_inicial = 10
e = 0.005
p = 0.5
p1 = 0.1 - e
p2 = 0.75 - e

# Creamos el grafo de mundo pequeño usando el modelo de Watts-Strogatz
k = 10 # Cada nodo se conecta a k vecinos
p_ws = 0.1 # Probabilidad de reconexión
G = nx.watts_strogatz_graph(numero_jugadores, k, p_ws)

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if random.random() < p:
            vecinos = conseguir_vecinos(i, numero_jugadores)
            j = random.choice(vecinos)

            if capitales[i] >= capitales[j]:
                capitales[i] -= 1
                capitales[j] += 1
            else:
                capitales[i] += 1
                capitales[j] -= 1
        else:
            if capitales[i] % 3 == 0:
                if random.random() < p1:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if random.random() < p2:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1

    capital_total = sum(capitales)

    evolucion_media[iteracion] += capital_total / numero_simulaciones

print("Evolución media del capital total:")
print(evolucion_media)

iterations = list(range(1, len(evolucion_media) + 1))
plt.scatter(iterations, evolucion_media, color='blue')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()
```

7.1.22. Evolución del capital para una red de mundo pequeño con el patrón 4 de comportamiento

```
import matplotlib.pyplot as plt
import random
import networkx as nx
import numpy as np

numero_jugadores = 500
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 10
capital_inicial = 10
e = 0.005
p = 0.5
p1 = 0.1 - e
p2 = 0.75 - e

k = 10
p_ws = 0.1
G = nx.watts_strogatz_graph(numero_jugadores, k, p_ws)

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if random.random() < p:
            vecinos = list(G.neighbors(i))
            j = random.choice(vecinos)

            if capitales[i] >= capital_inicial:
                capitales[i] -= 1
                capitales[j] += 1
            else:
                if random.random() < 0.5:
                    capitales[i] += 1
                    capitales[j] -= 1
                else:
                    capitales[i] -= 1
                    capitales[j] += 1
            else:
                if capitales[i] % 3 == 0:
                    if random.random() < p1:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1
                else:
                    if random.random() < p2:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1

        capital_total = sum(capitales)

        evolucion_media[iteracion] += capital_total / numero_simulaciones

print("Evolución media del capital total:")
print(evolucion_media)

iterations = list(range(1, len(evolucion_media) + 1))
plt.scatter(iterations, evolucion_media, color='blue')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()
```

7.1.23. Evolución del capital separando en altruistas y egoistas

```
import random
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt

numero_jugadores = 50
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 100
capital_inicial = 10
e = 0.005
p = 0.5 # Probabilidad de que un agente altruista juegue al juego A
p1 = 0.1 - e
p2 = 0.75 - e

G = nx.watts_strogatz_graph(numero_jugadores, k=4, p=0.1)

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    # Asignación aleatoria del tipo de jugador
    tipos_jugador = ['altruista' if random.random() < 0.5 else 'egoista' for _ in range(numero_jugadores)]

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if tipos_jugador[i] == 'altruista':
            # Para jugadores altruistas, podemos tener el juego A o el B
            if random.random() < p:
                vecinos = list(G.neighbors(i))
                if vecinos: # Para asegurarnos de que no es nulo
                    j = random.choice(vecinos)
                    capitales[i] -= 1
                    capitales[j] += 1
            else:
                if capitales[i] % 3 == 0:
                    if random.random() < p1:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1
                else:
                    if random.random() < p2:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1
        else:
            # Juego B solo para jugadores egoistas
            if capitales[i] % 3 == 0:
                if random.random() < p1:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if random.random() < p2:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1

        # Calcular el capital total al final de esta iteración
        total_capital = sum(capitales)

        # Agregar el total_capital al vector de evolución media
        evolucion_media[iteracion] += total_capital / numero_simulaciones

    # Imprimir la evolución media del capital total
    print("Evolución media del capital total:")
    print(evolucion_media)
```

```

# Crear el gráfico de puntos
iterations = list(range(1, numero_iteraciones_por_simulacion + 1))
plt.scatter(iterations, evolucion_media, color='blue')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()

```

7.1.24. Comparativa del capital de los altruistas y los egoístas

```

import random
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt

numero_jugadores = 50
numero_iteraciones_por_simulacion = 20
numero_simulaciones = 100
capital_inicial = 10
e = 0.005
p = 0.5 # Probabilidad de que un agente altruista juegue al juego A
p1 = 0.1 - e
p2 = 0.75 - e

G = nx.watts_strogatz_graph(numero_jugadores, k=4, p=0.1)

total_altruista = 0
total_egoista = 0

for sim in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    tipos_jugador = ['altruista' if random.random() < 0.5 else 'egoista' for _ in range(numero_jugadores)]

    capital_total_altruista = 0
    capital_total_egoista = 0

    for iteracion in range(numero_iteraciones_por_simulacion):
        i = random.randint(0, numero_jugadores - 1)

        if tipos_jugador[i] == 'altruista':
            # Para jugadores altruistas, podemos tener el juego A o el B
            if random.random() < p:
                vecinos = list(G.neighbors(i))
                if vecinos: # Para asegurarnos de que no es nulo
                    j = random.choice(vecinos)
                    capitales[i] -= 1
                    capitales[j] += 1
                    capital_total_altruista -= 1
            else:
                if capitales[i] % 3 == 0:
                    if random.random() < p1:
                        capitales[i] += 1
                        capital_total_altruista += 1
                    else:
                        capitales[i] -= 1
                        capital_total_altruista -= 1
                else:
                    if random.random() < p2:
                        capitales[i] += 1
                        capital_total_altruista += 1
                    else:
                        capitales[i] -= 1
                        capital_total_altruista -= 1
        else:
            if capitales[i] % 3 == 0:
                if random.random() < p1:

```

```

        capitales[i] += 1
        capital_total_egoista += 1
    else:
        capitales[i] -= 1
        capital_total_egoista -= 1
else:
    if random.random() < p2:
        capitales[i] += 1
        capital_total_egoista += 1
    else:
        capitales[i] -= 1
        capital_total_egoista -= 1

total_altruista += capital_total_altruista
total_egoista += capital_total_egoista

print(total_altruista/(numero_iteraciones_por_simulacion*numero_simulaciones*numero_jugadores))
print(total_egoista/(numero_iteraciones_por_simulacion*numero_simulaciones*numero_jugadores))

```

7.1.25. Caso síncrono democrático

```

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for simulacion in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        votes = ['A' if capitales[i] % M == 0 else 'B' for i in range(numero_jugadores)]

        if votes.count('A') > votes.count('B'):
            juego_elegido = 'A'
        else:
            juego_elegido = 'B'

        for i in range(numero_jugadores):
            if juego_elegido == 'A':
                if random.random() < p:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if capitales[i] % M == 0:
                    if random.random() < p1:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1
                else:
                    if random.random() < p2:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1

        capital_total = sum(capitales)

        evolucion_media[iteracion] += capital_total / numero_simulaciones

iteraciones = list(range(1, numero_iteraciones_por_simulacion + 1))
plt.plot(iteraciones, evolucion_media, marker='o', linestyle='-', color='b')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()

```

7.1.26. Caso síncrono aleatorio

```

import random
import numpy as np
import matplotlib.pyplot as plt

```

```

numero_jugadores = 500
numero_iteraciones_por_simulacion = 100
numero_simulaciones = 100
capital_inicial = 10
e = 0.005
M = 3
p = 0.5 - e
p1 = 0.1 - e
p2 = 0.75 - e

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for simulacion in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

    for iteracion in range(numero_iteraciones_por_simulacion):
        # Cada jugador vota por el juego que quiere jugar, de manera aleatoria
        votes = ['A' if random.random() < 0.5 else 'B' for _ in range(numero_jugadores)]

        # Decidir el juego elegido por la mayoría
        if votes.count('A') > votes.count('B'):
            juego_elegido = 'A'
        else:
            juego_elegido = 'B'

        # Jugar el juego elegido por la mayoría
        for i in range(numero_jugadores):
            if juego_elegido == 'A':
                if random.random() < p:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if capitales[i] % M == 0:
                    if random.random() < p1:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1
                else:
                    if random.random() < p2:
                        capitales[i] += 1
                    else:
                        capitales[i] -= 1

        # Calculamos el capital total al final de esta iteración
        capital_total = sum(capitales)

        # Calculamos evolución media
        evolucion_media[iteracion] += capital_total / numero_simulaciones

# Creamos el gráfico de puntos
iteraciones = list(range(1, numero_iteraciones_por_simulacion + 1))
plt.plot(iteraciones, evolucion_media, marker='o', linestyle='-', color='b')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()

```

7.1.27. Caso síncrono con dictador

```

dictador = random.randint(0, numero_jugadores - 1)

evolucion_media = np.zeros(numero_iteraciones_por_simulacion)

for simulacion in range(numero_simulaciones):
    capitales = [capital_inicial] * numero_jugadores

```

```

for iteracion in range(numero_iteraciones_por_simulacion):
    if capitales[dictador] % M == 0:
        votes = "A"
    else:
        votes = "B"

    if votes.count('A') > votes.count('B'):
        juego_elegido = 'A'
    else:
        juego_elegido = 'B'

    for i in range(numero_jugadores):
        if juego_elegido == 'A':
            if random.random() < p:
                capitales[i] += 1
            else:
                capitales[i] -= 1
        else:
            if capitales[i] % M == 0:
                if random.random() < p1:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1
            else:
                if random.random() < p2:
                    capitales[i] += 1
                else:
                    capitales[i] -= 1

    capital_total = sum(capitales)

    evolucion_media[iteracion] += capital_total / numero_simulaciones

iteraciones = list(range(1, numero_iteraciones_por_simulacion + 1))
plt.plot(iteraciones, evolucion_media, marker='o', linestyle='-', color='b')
plt.title('Evolución Media del Capital Total')
plt.xlabel('Iteraciones')
plt.ylabel('Capital Total Medio')
plt.grid(True)
plt.show()

print("Capital del dictador:")
print(capitales[dictador])
capital_sin_dictador = [capital for i, capital in enumerate(capitales) if i != dictador]
capital_medio_resto = sum(capital_sin_dictador) / (numero_jugadores - 1)
print("Capital medio del resto de jugadores:")
print(capital_medio_resto)

```

7.1.28. Convergencia de diferentes casos de juegos

Comprobemos la convergencia del juego A B B. Para ello, lo haremos de dos maneras. Primero, daremos valores aleatorios a las probabilidades, # variando el valor de epsilon en cada ejecución. Veremos que, aunque las # matrices A y B en cada ejecución no son iguales, la matriz A B B elevado a un # número muy grande (en este caso 200) da una matriz en la que los valores # en la misma columna son los mismos. Esto quiere decir que, para cualquier # distribución inicial, se llega a la distribución límite dada por esos valores # que se repiten en la matriz, pues todas las distribuciones suman 1. # Obviamente, esta distribución límite depende de los valores de las # probabilidades en cada ejecución

```

import numpy as np
import random

epsilon = random.random() * 0.1

A = np.array([
    [0, 0.5-epsilon, 0.5+epsilon],

```

```

        [0.5+epsilon, 0, 0.5-epsilon],
        [0.5-epsilon, 0.5+epsilon, 0]
    ])

    B = np.array([
        [0, 0.1-epsilon, 0.9+epsilon],
        [0.25+epsilon, 0, 0.75-epsilon],
        [0.75-epsilon, 0.25+epsilon, 0]
    ])

    ABB = A@B@B

    ABB = np.linalg.matrix_power(ABB, 200)

    ABB

    # Ahora lo que haremos será fijar las probabilidades que se emplean
    # en las matrices, y en cada ejecución crear una distribución inicial nueva
    # de manera aleatoria. Veremos que aunque la distribución inicial de la que
    # se parte va variando, todas acaban convergiendo a la misma

    epsilon = 0

    A = np.array([
        [0, 0.5-epsilon, 0.5+epsilon],
        [0.5+epsilon, 0, 0.5-epsilon],
        [0.5-epsilon, 0.5+epsilon, 0]
    ])

    B = np.array([
        [0, 0.1-epsilon, 0.9+epsilon],
        [0.25+epsilon, 0, 0.75-epsilon],
        [0.75-epsilon, 0.25+epsilon, 0]
    ])

    ABB = A@B@B

    vector_aleatorio = np.random.rand(3)

    # Normalizamos el vector
    vector_probabilidades = vector_aleatorio / np.sum(vector_aleatorio)

    print(vector_probabilidades)

    for i in range(200):
        vector_probabilidades = np.dot(vector_probabilidades, ABB)

    print(vector_probabilidades)

    # Implementamos las dos maneras para más casos ahora para la
    # segunda variante del juego A cooperativo
    # (aquí, por construcción de la matriz, las probabilidades
    # en ella son siempre las mismas)

    A = np.array([
        [0, 1/3, 1/3, 0, 1/3, 0, 0, 0],
        [0, 1/3, 1/6, 1/6, 1/6, 1/6, 0, 0],
        [0, 1/6, 1/3, 1/6, 1/6, 0, 1/6, 0],
        [0, 1/6, 1/6, 1/3, 0, 1/6, 1/6, 0],
        [0, 1/6, 1/6, 0, 1/3, 1/6, 1/6, 0],
        [0, 1/6, 0, 1/6, 1/6, 1/3, 1/6, 0],
        [0, 0, 1/6, 1/6, 1/6, 1/6, 1/3, 0],
        [0, 0, 0, 1/3, 0, 1/3, 1/3, 0]
    ])

    A_200 = np.linalg.matrix_power(A, 200)

    A_200

    vector_aleatorio = np.random.rand(8)

```

```

vector_probabilidades = vector_aleatorio / np.sum(vector_aleatorio)

print(vector_probabilidades)

for i in range(200):
    vector_probabilidades = np.dot(vector_probabilidades, A)

print(vector_probabilidades)

# Ahora para el juego A en la mezcla de 3

p=random.random()

A = np.array([
    [0, 0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0],
    [1 - p, 0, 0, 0, 0, 0, p, 0, 0, 0, 0, 0],
    [0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p],
    [0, 0, 0, 0, 0, 0, 1 - p, 0, 0, 0, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0],
    [0, 0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0],
    [1 - p, 0, 0, 0, 0, 0, p, 0, 0, 0, 0, 0],
    [0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0],
    [0, 0, 0, 0, 0, 0, 1 - p, 0, 0, 0, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0]
])

A_200 = np.linalg.matrix_power(A, 200)

A_200

vector_aleatorio = np.random.rand(12)

vector_probabilidades = vector_aleatorio / np.sum(vector_aleatorio)

print(vector_probabilidades)

p = 0.5
A = np.array([
    [0, 0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0],
    [1 - p, 0, 0, 0, 0, 0, p, 0, 0, 0, 0, 0, 0],
    [0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0],
    [0, 0, 0, 0, 0, 0, 1 - p, 0, 0, 0, 0, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0, 0],
    [0, 0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0],
    [1 - p, 0, 0, 0, 0, 0, p, 0, 0, 0, 0, 0, 0],
    [0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0],
    [0, 0, 0, 0, 0, 0, 1 - p, 0, 0, 0, 0, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0, 0]
])

for i in range(200):
    vector_probabilidades = np.dot(vector_probabilidades, A)

print(vector_probabilidades)

# Juego B en la mezcla de 3

pb=random.random()
pm=random.random()

B = np.array([
    [0, 0, 1 - pm, 0, pm, 0, 0, 0, 0, 0, 0, 0, 0],
    [1 - pb, 0, 0, 0, 0, 0, pb, 0, 0, 0, 0, 0, 0],
    [0, 1 - pb, 0, pb, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - pm, 0, pm, 0],

```

```

[0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, 0, 0, 0, pb],
[0, 0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, pb, 0, 0],
[0, 0, 1 - pm, 0, pm, 0, 0, 0, 0, 0, 0, 0, 0],
[1 - pb, 0, 0, 0, 0, 0, pb, 0, 0, 0, 0, 0, 0],
[0, 1 - pb, 0, pb, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - pm, 0, pm, 0],
[0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, 0, 0, 0, pb],
[0, 0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, pb, 0, 0]
])

B_200 = np.linalg.matrix_power(B, 200)

B_200

vector_aleatorio = np.random.rand(12)

vector_probabilidades = vector_aleatorio / np.sum(vector_aleatorio)

print(vector_probabilidades)

pm = 0.1
pb = 0.75
B = np.array([
    [0, 0, 1 - pm, 0, pm, 0, 0, 0, 0, 0, 0, 0, 0],
    [1 - pb, 0, 0, 0, 0, 0, pb, 0, 0, 0, 0, 0, 0],
    [0, 1 - pb, 0, pb, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - pm, 0, pm, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, 0, 0, 0, pb],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, pb, 0, 0],
    [0, 0, 1 - pm, 0, pm, 0, 0, 0, 0, 0, 0, 0, 0],
    [1 - pb, 0, 0, 0, 0, 0, pb, 0, 0, 0, 0, 0, 0],
    [0, 1 - pb, 0, pb, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - pm, 0, pm, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, 0, 0, 0, pb],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, pb, 0, 0]
])

for i in range(200):
    vector_probabilidades = np.dot(vector_probabilidades, B)

print(vector_probabilidades)

# Juego C en la mezcla de 3

p1=random.random()
p2=random.random()
p3=random.random()
p4=random.random()

C = np.array([
    [0, 0, 1 - p1, 0, p1, 0, 0, 0, 0, 0, 0, 0, 0],
    [1 - p1, 0, 0, 0, 0, 0, p1, 0, 0, 0, 0, 0, 0],
    [0, 1 - p1, 0, p1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, p2, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, 0, 0, 0, p2],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, p2, 0, 0],
    [0, 0, 1 - p3, 0, p3, 0, 0, 0, 0, 0, 0, 0, 0],
    [1 - p3, 0, 0, 0, 0, 0, p3, 0, 0, 0, 0, 0, 0],
    [0, 1 - p3, 0, p3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, p4, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, 0, 0, 0, p4],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, p4, 0, 0]
])

C_200 = np.linalg.matrix_power(C, 200)

C_200

vector_aleatorio = np.random.rand(12)

```

```

vector_probabilidades = vector_aleatorio / np.sum(vector_aleatorio)

print(vector_probabilidades)

p1 = 1/4
p2 = 6/10
p3 = 6/10
p4 = 5/8

C = np.array([
    [0, 0, 1 - p1, 0, p1, 0, 0, 0, 0, 0, 0, 0],
    [1 - p1, 0, 0, 0, 0, 0, p1, 0, 0, 0, 0, 0],
    [0, 1 - p1, 0, p1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, p2, 0],
    [0, 0, 0, 0, 0, 0, 1 - p2, 0, 0, 0, 0, p2],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, p2, 0],
    [0, 0, 1 - p3, 0, p3, 0, 0, 0, 0, 0, 0, 0],
    [1 - p3, 0, 0, 0, 0, 0, p3, 0, 0, 0, 0, 0],
    [0, 1 - p3, 0, p3, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, p4],
    [0, 0, 0, 0, 0, 0, 1 - p4, 0, 0, 0, 0, p4],
    [0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, p4, 0, 0]
])

for i in range(200):
    vector_probabilidades = np.dot(vector_probabilidades, C)

print(vector_probabilidades)

# Juego A+B+C en la mezcla de 3

p=random.random()
pb=random.random()
pm=random.random()
p1=random.random()
p2=random.random()
p3=random.random()
p4=random.random()

A = np.array([
    [0, 0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0],
    [1 - p, 0, 0, 0, 0, 0, p, 0, 0, 0, 0, 0],
    [0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, 0, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0],
    [0, 0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0],
    [1 - p, 0, 0, 0, 0, 0, p, 0, 0, 0, 0, 0],
    [0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0],
    [0, 0, 0, 0, 0, 0, 1 - p, 0, 0, 0, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0]
])

B = np.array([
    [0, 0, 1 - pm, 0, pm, 0, 0, 0, 0, 0, 0, 0],
    [1 - pb, 0, 0, 0, 0, 0, pb, 0, 0, 0, 0, 0],
    [0, 1 - pb, 0, pb, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - pm, 0, pm, 0],
    [0, 0, 0, 0, 0, 0, 1 - pb, 0, 0, 0, 0, pb],
    [0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, pb, 0, 0],
    [0, 0, 1 - pm, 0, pm, 0, 0, 0, 0, 0, 0, 0],
    [1 - pb, 0, 0, 0, 0, 0, pb, 0, 0, 0, 0, 0],
    [0, 1 - pb, 0, pb, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - pm, 0, pm, 0],
    [0, 0, 0, 0, 0, 0, 1 - pb, 0, 0, 0, 0, pb],
    [0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, pb, 0, 0]
])

```

```

C = np.array([
    [0, 0, 1 - p1, 0, p1, 0, 0, 0, 0, 0, 0, 0],
    [1 - p1, 0, 0, 0, 0, p1, 0, 0, 0, 0, 0, 0],
    [0, 1 - p1, 0, p1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, p2, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, 0, 0, p2],
    [0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, p2, 0, 0],
    [0, 0, 1 - p3, 0, p3, 0, 0, 0, 0, 0, 0, 0],
    [1 - p3, 0, 0, 0, 0, p3, 0, 0, 0, 0, 0, 0],
    [0, 1 - p3, 0, p3, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, p4, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, 0, 0, p4],
    [0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, p4, 0, 0]
])

F=(1/3)*(A+B+C)

F_200 = np.linalg.matrix_power(F, 200)

F_200

vector_aleatorio = np.random.rand(12)

vector_probabilidades = vector_aleatorio / np.sum(vector_aleatorio)

print(vector_probabilidades)

p = 0.5
pm = 0.1
pb = 0.75
p1 = 1/4
p2 = 6/10
p3 = 6/10
p4 = 5/8

A = np.array([
    [0, 0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0],
    [1 - p, 0, 0, 0, 0, p, 0, 0, 0, 0, 0, 0],
    [0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, 0, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0],
    [0, 0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0],
    [1 - p, 0, 0, 0, 0, p, 0, 0, 0, 0, 0, 0],
    [0, 1 - p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, 0, 0, p],
    [0, 0, 0, 0, 0, 0, 0, 1 - p, 0, p, 0, 0]
])

B = np.array([
    [0, 0, 1 - pm, 0, pm, 0, 0, 0, 0, 0, 0, 0],
    [1 - pb, 0, 0, 0, 0, pb, 0, 0, 0, 0, 0, 0],
    [0, 1 - pb, 0, pb, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - pm, 0, pm, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, 0, 0, pb],
    [0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, pb, 0, 0],
    [0, 0, 1 - pm, 0, pm, 0, 0, 0, 0, 0, 0, 0],
    [1 - pb, 0, 0, 0, 0, pb, 0, 0, 0, 0, 0, 0],
    [0, 1 - pb, 0, pb, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1 - pm, 0, pm],
    [0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, 0, 0, pb],
    [0, 0, 0, 0, 0, 0, 0, 1 - pb, 0, pb, 0, 0]
])

C = np.array([
    [0, 0, 1 - p1, 0, p1, 0, 0, 0, 0, 0, 0, 0],
    [1 - p1, 0, 0, 0, 0, p1, 0, 0, 0, 0, 0, 0],
    [0, 1 - p1, 0, p1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, p2, 0],

```

```

    [0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, 0, 0, 0, p2],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p2, 0, p2, 0, 0],
    [0, 0, 1 - p3, 0, p3, 0, 0, 0, 0, 0, 0, 0, 0],
    [1 - p3, 0, 0, 0, 0, 0, p3, 0, 0, 0, 0, 0, 0],
    [0, 1 - p3, 0, p3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, p4, 0],
    [0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, 0, 0, 0, p4],
    [0, 0, 0, 0, 0, 0, 0, 1 - p4, 0, p4, 0, 0, 0]
])

F=(1/3)*(A+B+C)

for i in range(200):
    vector_probabilidades = np.dot(vector_probabilidades, F)

print(vector_probabilidades)

# Comprobemos si se verifica para el juego B original
# con M par, por ejemplo M=4

epsilon = random.random() * 0.1

B = np.array([
    [0, 0.1-epsilon, 0, 0.9+epsilon],
    [0.25+epsilon, 0, 0.75-epsilon, 0],
    [0, 0.25+epsilon, 0, 0.75-epsilon],
    [0.75-epsilon, 0, 0.25+epsilon, 0],
])

B_200 = np.linalg.matrix_power(B, 200)

B_200

# Parece que en esta matriz no se puede asegurar la existencia de distribución
# estacionaria que coincida con la límite

# Probemos la segunda manera

epsilon = 0

B = np.array([
    [0, 0.1-epsilon, 0, 0.9+epsilon],
    [0.25+epsilon, 0, 0.75-epsilon, 0],
    [0, 0.25+epsilon, 0, 0.75-epsilon],
    [0.75-epsilon, 0, 0.25+epsilon, 0],
])

vector_aleatorio = np.random.rand(4)

vector_probabilidades = vector_aleatorio / np.sum(vector_aleatorio)

print(vector_probabilidades)

for i in range(200):
    vector_probabilidades = np.dot(vector_probabilidades, B)

print(vector_probabilidades)

# Efectivamente con este código no se llega en cada iteración
# a la misma distribución

```

7.2. Anexo 2: Códigos de *Mathematica*

7.2.1. Distribuciones estacionarias en los juegos originales

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego A en los juegos originales, con $\epsilon = 0$

```
In[ ]:=  $\epsilon = 0$ ;
```

```
In[ ]:=  $p = 0.5 - \epsilon$ ;
```

```
In[ ]:=  $A = \begin{pmatrix} 0 & p & 1-p \\ 1-p & 0 & p \\ p & 1-p & 0 \end{pmatrix}$ ;
```

```
In[ ]:= Id = IdentityMatrix[3];  
[matriz identidad]
```

Con NullSpace nos devuelve los vectores que, multiplicados por una cierta matriz por la izquierda, van a parar al 0. Por lo tanto, con NullSpace[Id - Transpose[A]] obtenemos los autovectores de valor propio 1 (se transpone la matriz A porque en el trabajo la multiplicación es por la derecha)

```
In[ ]:= autovectores = NullSpace[Id - Transpose[A]];  
[espacio nulo] [transposición]
```

```
In[ ]:= autovectoresNormalizados =  
autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])
```

```
Out[ ]:=  
{ {0.333333, 0.333333, 0.333333} }
```

Observamos que la distribución estacionaria es (1/3, 1/3, 1/3)

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego B en los juegos originales, con $\epsilon = 0$

```
In[ ]:=  $pm = 1 / 10 - \epsilon$ ;
```

```
 $pb = 3 / 4 - \epsilon$ ;
```

```
In[ ]:=  $B = \begin{pmatrix} 0 & pm & 1-pm \\ 1-pb & 0 & pb \\ pb & 1-pb & 0 \end{pmatrix}$ ;
```

```
In[ ]:= autovectores = NullSpace[Id - Transpose[B]];  
[espacio nulo] [transposición]
```

```
In[ ]:= autovectoresNormalizados =  
autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])
```

```
Out[ ]:=  
{ { $\frac{5}{13}$ ,  $\frac{2}{13}$ ,  $\frac{6}{13}$ } }
```

Cálculo de la distribución estacionaria y la probabilidad

de ganar del juego A en los juegos originales, sin determinar el valor de ϵ

```
In[ ]:= Clear[ϵ]
      borra

In[ ]:= p = 0.5 - ϵ;

In[ ]:= A =  $\begin{pmatrix} 0 & p & 1-p \\ 1-p & 0 & p \\ p & 1-p & 0 \end{pmatrix}$ ;

In[ ]:= autovectores = NullSpace[Id - Transpose[A]];
      espacio nulo      transposición

In[ ]:= autovectoresNormalizados =
      autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])

Out[ ]:= { {0.333333, 0.333333, 0.333333} }

Obtenemos de nuevo (1/3, 1/3, 1/3)

In[ ]:= Simplify[(autovectoresNormalizados[[1, 1]]) * (1 / 2 - ϵ) +
      simplifica
      (1 - autovectoresNormalizados[[1, 1]]) * (1 / 2 - ϵ)]

Out[ ]:= 0.5 - 1. ϵ
```

Esta es la probabilidad de ganar en A. Observamos que efectivamente el juego A es justo cuando ϵ vale 0 y perdedor si ϵ es mayor que 0

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego B en los juegos originales, sin determinar el valor de ϵ

```
In[ ]:= pm = 1 / 10 - ϵ;
      pb = 3 / 4 - ϵ;

In[ ]:= B =  $\begin{pmatrix} 0 & pm & 1-pm \\ 1-pb & 0 & pb \\ pb & 1-pb & 0 \end{pmatrix}$ ;

In[ ]:= autovectores = NullSpace[Id - Transpose[B]];
      espacio nulo      transposición
```

```
In[*]:= autovectoresNormalizados =  
autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])
```

$$\text{Out}[*]= \left\{ \frac{5 (13 - 8 \epsilon + 16 \epsilon^2)}{2 (39 + 6 \epsilon + 40 \epsilon^2) \left(1 - \frac{-13 - 6 \epsilon - 40 \epsilon^2}{39 + 6 \epsilon + 40 \epsilon^2} + \frac{5 (13 - 8 \epsilon + 16 \epsilon^2)}{2 (39 + 6 \epsilon + 40 \epsilon^2)} \right)}, \right. \\ \left. - \frac{-13 - 6 \epsilon - 40 \epsilon^2}{(39 + 6 \epsilon + 40 \epsilon^2) \left(1 - \frac{-13 - 6 \epsilon - 40 \epsilon^2}{39 + 6 \epsilon + 40 \epsilon^2} + \frac{5 (13 - 8 \epsilon + 16 \epsilon^2)}{2 (39 + 6 \epsilon + 40 \epsilon^2)} \right)}, \frac{1}{1 - \frac{-13 - 6 \epsilon - 40 \epsilon^2}{39 + 6 \epsilon + 40 \epsilon^2} + \frac{5 (13 - 8 \epsilon + 16 \epsilon^2)}{2 (39 + 6 \epsilon + 40 \epsilon^2)}} \right\}$$

Como la expresión de la distribución estacionaria es muy larga, hacemos el desarrollo de Taylor de orden 2 de cada componente

```
In[*]:= Series[autovectoresNormalizados[[1, 1]], {\epsilon, 0, 2}]  
[serie]
```

$$\text{Out}[*]= \frac{5}{13} - \frac{440 \epsilon}{2197} - \frac{34080 \epsilon^2}{371293} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 2]], {\epsilon, 0, 2}]  
[serie]
```

$$\text{Out}[*]= \frac{2}{13} + \frac{188 \epsilon}{2197} + \frac{97648 \epsilon^2}{371293} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 3]], {\epsilon, 0, 2}]  
[serie]
```

$$\text{Out}[*]= \frac{6}{13} + \frac{252 \epsilon}{2197} - \frac{63568 \epsilon^2}{371293} + O[\epsilon]^3$$

```
In[*]:= (autovectoresNormalizados[[1, 1]]) * (0.1 - \epsilon) +  
(1 - autovectoresNormalizados[[1, 1]]) * (0.75 - \epsilon)
```

$$\text{Out}[*]= \frac{5 (0.1 - \epsilon) (13 - 8 \epsilon + 16 \epsilon^2)}{2 (39 + 6 \epsilon + 40 \epsilon^2) \left(1 - \frac{-13 - 6 \epsilon - 40 \epsilon^2}{39 + 6 \epsilon + 40 \epsilon^2} + \frac{5 (13 - 8 \epsilon + 16 \epsilon^2)}{2 (39 + 6 \epsilon + 40 \epsilon^2)} \right)} + \\ (0.75 - \epsilon) \left(1 - \frac{5 (13 - 8 \epsilon + 16 \epsilon^2)}{2 (39 + 6 \epsilon + 40 \epsilon^2) \left(1 - \frac{-13 - 6 \epsilon - 40 \epsilon^2}{39 + 6 \epsilon + 40 \epsilon^2} + \frac{5 (13 - 8 \epsilon + 16 \epsilon^2)}{2 (39 + 6 \epsilon + 40 \epsilon^2)} \right)} \right)$$

Como la probabilidad viene dada por una expresión muy larga, hacemos su desarrollo de Taylor en $\epsilon = 0$

```
In[*]:= Series[(autovectoresNormalizados[[1, 1]]) * (0.1 - \epsilon) +  
[serie]  
(1 - autovectoresNormalizados[[1, 1]]) * (0.75 - \epsilon), {\epsilon, 0, 2}]
```

$$\text{Out}[*]= 0.5 - 0.869822 \epsilon + 0.0596618 \epsilon^2 + O[\epsilon]^3$$

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego A+B en los juegos originales, sin determinar el valor de ϵ

```

In[*]:= matrizAleatoria = (1 / 2) * (A + B);

In[*]:= autovectores = NullSpace[Id - Transpose[matrizAleatoria]];
|espacio nulo |transposición

In[*]:= autovectoresNormalizados =
autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])

Out[*]:=

$$\left\{ \left\{ \frac{1. (0.765625 - 0.25 \epsilon + 1. \epsilon^2)}{(0.8875 + 0.075 \epsilon + 1. \epsilon^2) \left( 1. + \frac{1. (0.765625 - 0.25 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} + \frac{1. (0.5625 + 0.075 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} \right)}, \right. \right.$$


$$\frac{1. (0.5625 + 0.075 \epsilon + 1. \epsilon^2)}{(0.8875 + 0.075 \epsilon + 1. \epsilon^2) \left( 1. + \frac{1. (0.765625 - 0.25 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} + \frac{1. (0.5625 + 0.075 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} \right)},$$


$$\left. \left. \frac{1.}{1. + \frac{1. (0.765625 - 0.25 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} + \frac{1. (0.5625 + 0.075 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2}} \right\} \right\}$$


In[*]:= Series[autovectoresNormalizados[[1, 1]], { $\epsilon$ , 0, 2}]
|serie

Out[*]:=
 $0.345557 - 0.0972386 \epsilon - 0.02094 \epsilon^2 + 0 [\epsilon]^3$ 

In[*]:= Series[autovectoresNormalizados[[1, 2]], { $\epsilon$ , 0, 2}]
|serie

Out[*]:=
 $0.253879 + 0.0453091 \epsilon + 0.109628 \epsilon^2 + 0 [\epsilon]^3$ 

In[*]:= Series[autovectoresNormalizados[[1, 3]], { $\epsilon$ , 0, 2}]
|serie

Out[*]:=
 $0.400564 + 0.0519296 \epsilon - 0.0886881 \epsilon^2 + 0 [\epsilon]^3$ 

In[*]:= (1 / 2) (1 / 2 -  $\epsilon$ ) + (1 / 2) * ((autovectoresNormalizados[[1, 1]]) * (1 / 10 -  $\epsilon$ ) +
(1 - (autovectoresNormalizados[[1, 1]])) * ((3 / 4) -  $\epsilon$ ))

Out[*]:=

$$\frac{1}{2} \left( \frac{1}{2} - \epsilon \right) + \frac{1}{2} \left( \frac{1. \left( \frac{1}{10} - \epsilon \right) (0.765625 - 0.25 \epsilon + 1. \epsilon^2)}{(0.8875 + 0.075 \epsilon + 1. \epsilon^2) \left( 1. + \frac{1. (0.765625 - 0.25 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} + \frac{1. (0.5625 + 0.075 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} \right)} + \right.$$


$$\left. \left( \frac{3}{4} - \epsilon \right) \left( 1 - \frac{1. (0.765625 - 0.25 \epsilon + 1. \epsilon^2)}{(0.8875 + 0.075 \epsilon + 1. \epsilon^2) \left( 1. + \frac{1. (0.765625 - 0.25 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} + \frac{1. (0.5625 + 0.075 \epsilon + 1. \epsilon^2)}{0.8875 + 0.075 \epsilon + 1. \epsilon^2} \right)} \right) \right)$$


```

```
In[*]:= Series[(1/2) (1/2 - ε) + (1/2) * ((autovectoresNormalizados[[1, 1]]) * (1/10 - ε) +
[serie
(1 - (autovectoresNormalizados[[1, 1]])) * ((3/4) - ε)), {ε, 0, 2}]
```

```
Out[*]= 0.512694 - 0.968397 ε + 0.00680551 ε^2 + O[ε]^3
```

Efectivamente, la probabilidad es mayor que 0.5 para $\epsilon = 0$

¿ Hasta qué valor de ϵ se verifica la paradoja ?

```
In[*]:= NSolve[(1/2) (1/2 - ε) + (1/2) * ((autovectoresNormalizados[[1, 1]]) * (1/10 - ε) +
[resuelve numéricamente
(1 - (autovectoresNormalizados[[1, 1]])) * ((3/4) - ε)) == 0.5, ε]
```

```
Out[*]= {{ε -> 0.0184454 - 0.845459 I}, {ε -> 0.0184454 + 0.845459 I}, {ε -> 0.0131093}}
```

7.2.2. Distribuciones estacionarias en los juegos dependientes de la historia

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego A en los juegos dependientes de la historia

```

In[ ]:= p = 1 / 2 - ε;

In[ ]:= A = 
$$\begin{pmatrix} 1-p & p & 0 & 0 \\ 0 & 0 & 1-p & p \\ 1-p & p & 0 & 0 \\ 0 & 0 & 1-p & p \end{pmatrix};$$


In[ ]:= Id = IdentityMatrix[4];
      matriz identidad

In[ ]:= autovectores = NullSpace[Id - Transpose[A]];
      espacio nulo      transposición

In[ ]:= autovectoresNormalizados = autovectores /
      (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]])

Out[ ]:=

$$\left\{ \left\{ \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2 \left(1 - \frac{2(1+2\epsilon)}{-1+2\epsilon} + \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2}\right)}, -\frac{1+2\epsilon}{(-1+2\epsilon) \left(1 - \frac{2(1+2\epsilon)}{-1+2\epsilon} + \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2}\right)}, \right. \right.$$


$$\left. -\frac{1+2\epsilon}{(-1+2\epsilon) \left(1 - \frac{2(1+2\epsilon)}{-1+2\epsilon} + \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2}\right)}, \frac{1}{1 - \frac{2(1+2\epsilon)}{-1+2\epsilon} + \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2}} \right\} \}$$


In[ ]:= Series[autovectoresNormalizados[[1, 1]], {ε, 0, 2}]
      serie

Out[ ]:=

$$\frac{1}{4} + \epsilon + \epsilon^2 + 0[\epsilon]^3$$


In[ ]:= Series[autovectoresNormalizados[[1, 2]], {ε, 0, 2}]
      serie

Out[ ]:=

$$\frac{1}{4} - \epsilon^2 + 0[\epsilon]^3$$


In[ ]:= Series[autovectoresNormalizados[[1, 3]], {ε, 0, 2}]
      serie

Out[ ]:=

$$\frac{1}{4} - \epsilon^2 + 0[\epsilon]^3$$


In[ ]:= Series[autovectoresNormalizados[[1, 4]], {ε, 0, 2}]
      serie

Out[ ]:=

$$\frac{1}{4} - \epsilon + \epsilon^2 + 0[\epsilon]^3$$


```

Observamos que para $\epsilon = 0$ la distribución estacionaria es (1/4, 1/4, 1/4, 1/4)

```
In[*]:= (autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 2]] +
          autovectoresNormalizados[[1, 3]] + autovectoresNormalizados[[1, 4]]) * (1 / 2 - ε)
```

```
Out[*]=
```

$$\left(\frac{1}{2} - \epsilon \right) \left(\frac{1}{1 - \frac{2(1+2\epsilon)}{-1+2\epsilon} + \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2}} - \frac{2(1+2\epsilon)}{(-1+2\epsilon) \left(1 - \frac{2(1+2\epsilon)}{-1+2\epsilon} + \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2} \right)} + \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2 \left(1 - \frac{2(1+2\epsilon)}{-1+2\epsilon} + \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2} \right)} \right)$$

```
In[*]:= Series[(autovectoresNormalizados[[1, 1]] +
                 autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]] +
                 autovectoresNormalizados[[1, 4]]) * (1 / 2 - ε), {ε, 0, 2}]
```

```
Out[*]=
```

$$\frac{1}{2} - \epsilon + 0[\epsilon]^3$$

Estamos ante un juego justo cuando $\epsilon = 0$

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego B' en los juegos dependientes de la historia

```
In[*]:= p1 = 9 / 10 - ε;
p2 = 1 / 4 - ε;
p3 = 1 / 4 - ε;
p4 = 7 / 10 - ε;
```

```
In[*]:= B =
```

$$\begin{pmatrix} 1-p1 & p1 & 0 & 0 \\ 0 & 0 & 1-p2 & p2 \\ 1-p3 & p3 & 0 & 0 \\ 0 & 0 & 1-p4 & p4 \end{pmatrix};$$

```
In[*]:= Id = IdentityMatrix[4];
```

```
In[*]:= autovectores = NullSpace[Id - Transpose[B]];
|espacio nulo |transposición
```

```
In[*]:= autovectoresNormalizados = autovectores /
          (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]])
```

```
Out[*]=
```

$$\left\{ \left\{ \frac{(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon) \left(1 - \frac{4(3+10\epsilon)}{5(-1+4\epsilon)} + \frac{(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon)} \right)}, \right. \right. \\ - \frac{2(3+10\epsilon)}{5(-1+4\epsilon) \left(1 - \frac{4(3+10\epsilon)}{5(-1+4\epsilon)} + \frac{(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon)} \right)}, \\ \left. \left. - \frac{2(3+10\epsilon)}{5(-1+4\epsilon) \left(1 - \frac{4(3+10\epsilon)}{5(-1+4\epsilon)} + \frac{(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon)} \right)}, \frac{1}{1 - \frac{4(3+10\epsilon)}{5(-1+4\epsilon)} + \frac{(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon)}} \right\} \right\}$$

```

In[ ]:= Series[autovectoresNormalizados[[1, 1]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{5}{22} + \frac{80 \epsilon}{99} + \frac{100 \epsilon^2}{891} + O[\epsilon]^3$$


In[ ]:= Series[autovectoresNormalizados[[1, 2]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{3}{11} + \frac{10 \epsilon}{33} - \frac{400 \epsilon^2}{297} + O[\epsilon]^3$$


In[ ]:= Series[autovectoresNormalizados[[1, 3]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{3}{11} + \frac{10 \epsilon}{33} - \frac{400 \epsilon^2}{297} + O[\epsilon]^3$$


In[ ]:= Series[autovectoresNormalizados[[1, 4]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{5}{22} - \frac{140 \epsilon}{99} + \frac{2300 \epsilon^2}{891} + O[\epsilon]^3$$


In[ ]:= (autovectoresNormalizados[[1, 1]] * (9 / 10 - ϵ) +
(autovectoresNormalizados[[1, 2]] * (1 / 4 - ϵ) +
(autovectoresNormalizados[[1, 3]] * (1 / 4 - ϵ) +
(autovectoresNormalizados[[1, 4]] * (7 / 10 - ϵ)
Out[ ]:=

$$\frac{\frac{7}{10} - \epsilon}{1 - \frac{4(3+10\epsilon)}{5(-1+4\epsilon)} + \frac{(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon)}} - \frac{4\left(\frac{1}{4} - \epsilon\right)(3+10\epsilon)}{5(-1+4\epsilon)\left(1 - \frac{4(3+10\epsilon)}{5(-1+4\epsilon)} + \frac{(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon)}\right)} +$$


$$\frac{\left(\frac{9}{10} - \epsilon\right)(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon)\left(1 - \frac{4(3+10\epsilon)}{5(-1+4\epsilon)} + \frac{(3+4\epsilon)(3+10\epsilon)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}$$


In[ ]:= Series[(autovectoresNormalizados[[1, 1]] * (9 / 10 - ϵ) +
(autovectoresNormalizados[[1, 2]] * (1 / 4 - ϵ) +
(autovectoresNormalizados[[1, 3]] * (1 / 4 - ϵ) +
(autovectoresNormalizados[[1, 4]] * (7 / 10 - ϵ)), {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{2} - \frac{10 \epsilon}{9} + \frac{100 \epsilon^2}{81} + O[\epsilon]^3$$


```

Efectivamente vemos que, cuando epsilon es 0, estamos ante un juego justo

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego A+B' en los juegos dependientes de la historia

In[]:= **P = (1 / 2) * (A + B);**

In[]:= **autovectores = NullSpace[Id - Transpose[P]];**
|espacio nulo |transposición

In[]:= **autovectoresNormalizados = autovectores /**
(autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]])

Out[]:=

$$\left\{ \left\{ \frac{2(2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon) \left(1 - \frac{16(2+5\epsilon)}{5(-3+8\epsilon)} + \frac{2(2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon)} \right)}, \right. \right. \\ \left. - \frac{8(2+5\epsilon)}{5(-3+8\epsilon) \left(1 - \frac{16(2+5\epsilon)}{5(-3+8\epsilon)} + \frac{2(2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon)} \right)}, \right. \\ \left. - \frac{8(2+5\epsilon)}{5(-3+8\epsilon) \left(1 - \frac{16(2+5\epsilon)}{5(-3+8\epsilon)} + \frac{2(2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon)} \right)}, \frac{1}{1 - \frac{16(2+5\epsilon)}{5(-3+8\epsilon)} + \frac{2(2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon)}} \right\} \right\}$$

In[]:= **Series[autovectoresNormalizados[[1, 1]], {ϵ, 0, 2}]**
|serie

Out[]:=

$$\frac{100}{429} + \frac{13990\epsilon}{16731} + \frac{328600\epsilon^2}{652509} + O[\epsilon]^3$$

In[]:= **Series[autovectoresNormalizados[[1, 2]], {ϵ, 0, 2}]**
|serie

Out[]:=

$$\frac{112}{429} + \frac{2440\epsilon}{16731} - \frac{657200\epsilon^2}{652509} + O[\epsilon]^3$$

In[]:= **Series[autovectoresNormalizados[[1, 3]], {ϵ, 0, 2}]**
|serie

Out[]:=

$$\frac{112}{429} + \frac{2440\epsilon}{16731} - \frac{657200\epsilon^2}{652509} + O[\epsilon]^3$$

In[]:= **Series[autovectoresNormalizados[[1, 4]], {ϵ, 0, 2}]**
|serie

Out[]:=

$$\frac{35}{143} - \frac{6290\epsilon}{5577} + \frac{328600\epsilon^2}{217503} + O[\epsilon]^3$$

In[]:= **(1 / 2) * (1 / 2 - ϵ) + (1 / 2) * (autovectoresNormalizados[[1, 1]] * (p1) +**
(autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]]) * (p2) +
(autovectoresNormalizados[[1, 4]]) * p4)

Out[]:=

$$\frac{1}{2} \left(\frac{1}{2} - \epsilon \right) + \frac{1}{2} \left(\frac{\frac{7}{10} - \epsilon}{1 - \frac{16(2+5\epsilon)}{5(-3+8\epsilon)} + \frac{2(2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon)}} - \frac{16 \left(\frac{1}{4} - \epsilon \right) (2+5\epsilon)}{5(-3+8\epsilon) \left(1 - \frac{16(2+5\epsilon)}{5(-3+8\epsilon)} + \frac{2(2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon)} \right)} + \right. \\ \left. \frac{2 \left(\frac{9}{10} - \epsilon \right) (2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon) \left(1 - \frac{16(2+5\epsilon)}{5(-3+8\epsilon)} + \frac{2(2+5\epsilon)(5+8\epsilon)}{(-3+8\epsilon)(-7+10\epsilon)} \right)} \right)$$

```
In[*]:= Series[(1/2)*(1/2 - ε) + (1/2)*((autovectoresNormalizados[[1, 1]])*(p1) +
[serie
      (autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]])*(p2) +
      (autovectoresNormalizados[[1, 4]])*p4), {ε, 0, 2}]
```

```
Out[*]=
      217   16430 ε   328600 ε2
      429   16731   652509   + O[ε]3
```

Efectivamente la probabilidad es mayor que 0.5 para un valor de ϵ lo suficientemente pequeño

¿ Hasta qué valor de ϵ se verifica la paradoja ?

```
In[*]:= NSolve[(1/2)*(1/2 - ε) + (1/2)*((autovectoresNormalizados[[1, 1]])*(p1) +
[resuelve numéricamente
      (autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]])*(p2) +
      (autovectoresNormalizados[[1, 4]])*p4) == 0.5, ε]
```

```
Out[*]=
      {{ε → 0.00595238}}
```

7.2.3. Distribuciones estacionarias en los juegos cooperativos en su primera variante

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego A en los juegos cooperativos

```
In[ ]:= p = 1 / 2 - ε;
```

```
In[ ]:= A =
```

$$\begin{pmatrix} 1-p & \frac{1}{3} * p & \frac{1}{3} * p & 0 & \frac{1}{3} \\ \frac{1}{3} * (1-p) & \frac{1}{3} * ((1-p) + (1-p) + p) & 0 & \frac{1}{3} * p & \\ \frac{1}{3} * (1-p) & 0 & \frac{1}{3} * ((1-p) + (1-p) + p) & \frac{1}{3} * p & \\ 0 & \frac{1}{3} * (1-p) & \frac{1}{3} * (1-p) & \frac{1}{3} * ((1-p) + p + p) & \frac{1}{3} * ((1-p) \\ \frac{1}{3} * (1-p) & 0 & 0 & 0 & \frac{1}{3} * ((1-p) \\ 0 & \frac{1}{3} * (1-p) & 0 & 0 & \frac{1}{3} * (\\ 0 & 0 & \frac{1}{3} * (1-p) & 0 & \frac{1}{3} * (\\ 0 & 0 & 0 & \frac{1}{3} * (1-p) & \end{pmatrix}$$

```
In[ ]:= Id = IdentityMatrix[8];
```

```
In[ ]:= autovectores = NullSpace[Id - Transpose[A]];
```

```
In[*]:= autovectoresNormalizados = autovectores /
      (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
      autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]])
```

$$\text{Out[*]} = \left\{ \left\{ -\frac{1 + 6\epsilon + 12\epsilon^2 + 8\epsilon^3}{(-1 + 2\epsilon)^3 \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)}, \right. \right. \\ \left. -\frac{-1 - 4\epsilon - 4\epsilon^2}{(-1 + 2\epsilon)^2 \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)}, \right. \\ \left. \frac{(1 + 2\epsilon)^2}{(-1 + 2\epsilon)^2 \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)}, \right. \\ \left. -\frac{1 + 2\epsilon}{(-1 + 2\epsilon) \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)}, \right. \\ \left. \frac{(1 + 2\epsilon)^2}{(-1 + 2\epsilon)^2 \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)}, \right. \\ \left. -\frac{1 + 2\epsilon}{(-1 + 2\epsilon) \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)}, \right. \\ \left. \frac{1}{(-1 + 2\epsilon) \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)} \right\} \right\} \\ 1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3}$$

```
In[*]:= Series[autovectoresNormalizados[[1, 1]], {\epsilon, 0, 2}]
      serie
```

$$\text{Out[*]} = \frac{1}{8} + \frac{3\epsilon}{4} + \frac{3\epsilon^2}{2} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 2]], {\epsilon, 0, 2}]
      serie
```

$$\text{Out[*]} = \frac{1}{8} + \frac{\epsilon}{4} - \frac{\epsilon^2}{2} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 3]], {\epsilon, 0, 2}]
      serie
```

$$\text{Out[*]} = \frac{1}{8} + \frac{\epsilon}{4} - \frac{\epsilon^2}{2} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 4]], {\epsilon, 0, 2}]
      serie
```

$$\text{Out[*]} = \frac{1}{8} - \frac{\epsilon}{4} - \frac{\epsilon^2}{2} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 5]], {ϵ, 0, 2}]
```

```
Out[*]=
```

$$\frac{1}{8} + \frac{\epsilon}{4} - \frac{\epsilon^2}{2} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 6]], {ϵ, 0, 2}]
```

```
Out[*]=
```

$$\frac{1}{8} - \frac{\epsilon}{4} - \frac{\epsilon^2}{2} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 7]], {ϵ, 0, 2}]
```

```
Out[*]=
```

$$\frac{1}{8} - \frac{\epsilon}{4} - \frac{\epsilon^2}{2} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 8]], {ϵ, 0, 2}]
```

```
Out[*]=
```

$$\frac{1}{8} - \frac{3\epsilon}{4} + \frac{3\epsilon^2}{2} + O[\epsilon]^3$$

Observamos que para $\epsilon = 0$ la distribución estacionaria es $(1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8)$

```
In[*]:= (autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 2]] +
autovectoresNormalizados[[1, 3]] + autovectoresNormalizados[[1, 4]] +
autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]] +
autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 8]]) * (1 / 2 - ϵ)
```

```
Out[*]=
```

$$\left(\frac{1}{2} - \epsilon \right) \left(\frac{1}{1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3}} - \frac{3(1+2\epsilon)}{(-1+2\epsilon) \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2 \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2 \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3 \left(1 - \frac{3(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} - \frac{1+6\epsilon+12\epsilon^2+8\epsilon^3}{(-1+2\epsilon)^3} \right)} \right)$$

```
In[ ]:= Series[(autovectoresNormalizados[[1, 1]] +
  serie
    autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]] +
    autovectoresNormalizados[[1, 4]] + autovectoresNormalizados[[1, 5]] +
    autovectoresNormalizados[[1, 6]] + autovectoresNormalizados[[1, 7]] +
    autovectoresNormalizados[[1, 8]]) * (1/2 - ε), {ε, 0, 2}]
```

```
Out[ ]:=

$$\frac{1}{2} - \epsilon + O[\epsilon]^3$$

```

Efectivamente, estamos ante un juego justo cuando $\epsilon = 0$

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego B'' en los juegos cooperativos

```
In[ ]:= p0 = a - ε;
p1 = b - ε;
p2 = b - ε;
p3 = c - ε;
```

```
In[ ]:= B =
```

$$\begin{pmatrix} 1 - p_0 & \frac{1}{3} * p_0 & \frac{1}{3} * p_0 & 0 & \frac{1}{3} * p_0 \\ \frac{1}{3} * (1 - p_0) & \frac{1}{3} * (2 - p_2 - p_1 + p_0) & 0 & \frac{1}{3} * p_1 & 0 \\ \frac{1}{3} * (1 - p_0) & 0 & \frac{1}{3} * (2 - p_2 - p_1 + p_0) & \frac{1}{3} * p_2 & 0 \\ 0 & \frac{1}{3} * (1 - p_1) & \frac{1}{3} * (1 - p_2) & \frac{1}{3} * (1 - p_3 + p_1 + p_2) & 0 \\ \frac{1}{3} * (1 - p_0) & 0 & 0 & 0 & \frac{1}{3} * (2 - p_2 - p_1 + p_0) \\ 0 & \frac{1}{3} * (1 - p_2) & 0 & 0 & \frac{1}{3} * (1 - p_1) \\ 0 & 0 & \frac{1}{3} * (1 - p_1) & 0 & \frac{1}{3} * (1 - p_2) \\ 0 & 0 & 0 & \frac{1}{3} * (1 - p_3) & 0 \end{pmatrix};$$

```
In[ ]:= autovectores = NullSpace[Id - Transpose[B]];
  [espacio nulo] [transposición]
```

```
In[ ]:= autovectoresNormalizados = autovectores /
  (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
    autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]]);
```

```

In[ ]:= (autovectoresNormalizados[[1, 1]]) *  $\left(\frac{1}{3} p_0 + \frac{1}{3} p_0 + \frac{1}{3} p_0\right) +$ 
      (autovectoresNormalizados[[1, 2]]) *  $\left(\frac{1}{3} p_2 + \frac{1}{3} p_1 + \frac{1}{3} p_0\right) +$ 
      (autovectoresNormalizados[[1, 3]]) *  $\left(\frac{1}{3} p_1 + \frac{1}{3} p_0 + \frac{1}{3} p_2\right) +$ 
      (autovectoresNormalizados[[1, 4]]) *  $\left(\frac{1}{3} p_3 + \frac{1}{3} p_1 + \frac{1}{3} p_2\right) +$ 
      (autovectoresNormalizados[[1, 5]]) *  $\left(\frac{1}{3} p_0 + \frac{1}{3} p_2 + \frac{1}{3} p_1\right) +$ 
      (autovectoresNormalizados[[1, 6]]) *  $\left(\frac{1}{3} p_2 + \frac{1}{3} p_3 + \frac{1}{3} p_1\right) +$ 
      (autovectoresNormalizados[[1, 7]]) *  $\left(\frac{1}{3} p_1 + \frac{1}{3} p_2 + \frac{1}{3} p_3\right) +$ 
      (autovectoresNormalizados[[1, 8]]) *  $\left(\frac{1}{3} p_3 + \frac{1}{3} p_3 + \frac{1}{3} p_3\right);$ 

In[ ]:= Simplify[ (autovectoresNormalizados[[1, 1]]) *  $\left(\frac{1}{3} p_0 + \frac{1}{3} p_0 + \frac{1}{3} p_0\right) +$ 
  |simplifica
      (autovectoresNormalizados[[1, 2]]) *  $\left(\frac{1}{3} p_2 + \frac{1}{3} p_1 + \frac{1}{3} p_0\right) +$ 
      (autovectoresNormalizados[[1, 3]]) *  $\left(\frac{1}{3} p_1 + \frac{1}{3} p_0 + \frac{1}{3} p_2\right) +$ 
      (autovectoresNormalizados[[1, 4]]) *  $\left(\frac{1}{3} p_3 + \frac{1}{3} p_1 + \frac{1}{3} p_2\right) +$ 
      (autovectoresNormalizados[[1, 5]]) *  $\left(\frac{1}{3} p_0 + \frac{1}{3} p_2 + \frac{1}{3} p_1\right) +$ 
      (autovectoresNormalizados[[1, 6]]) *  $\left(\frac{1}{3} p_2 + \frac{1}{3} p_3 + \frac{1}{3} p_1\right) +$ 
      (autovectoresNormalizados[[1, 7]]) *  $\left(\frac{1}{3} p_1 + \frac{1}{3} p_2 + \frac{1}{3} p_3\right) +$ 
      (autovectoresNormalizados[[1, 8]]) *  $\left(\frac{1}{3} p_3 + \frac{1}{3} p_3 + \frac{1}{3} p_3\right) ]$ 

Out[ ]:=

$$\frac{(1 + b - c) (a - \epsilon)}{1 - c + b (-1 + c - 2 \epsilon) + c \epsilon + a (2 + b - 2 c + \epsilon)}$$


```

```

In[*]:= Series[ (autovectoresNormalizados[[1, 1]]) *  $\left(\frac{1}{3} p_0 + \frac{1}{3} p_0 + \frac{1}{3} p_0\right) +$ 
|serie
    (autovectoresNormalizados[[1, 2]]) *  $\left(\frac{1}{3} p_2 + \frac{1}{3} p_1 + \frac{1}{3} p_0\right) +$ 
    (autovectoresNormalizados[[1, 3]]) *  $\left(\frac{1}{3} p_1 + \frac{1}{3} p_0 + \frac{1}{3} p_2\right) +$ 
    (autovectoresNormalizados[[1, 4]]) *  $\left(\frac{1}{3} p_3 + \frac{1}{3} p_1 + \frac{1}{3} p_2\right) +$ 
    (autovectoresNormalizados[[1, 5]]) *  $\left(\frac{1}{3} p_0 + \frac{1}{3} p_2 + \frac{1}{3} p_1\right) +$ 
    (autovectoresNormalizados[[1, 6]]) *  $\left(\frac{1}{3} p_2 + \frac{1}{3} p_3 + \frac{1}{3} p_1\right) +$ 
    (autovectoresNormalizados[[1, 7]]) *  $\left(\frac{1}{3} p_1 + \frac{1}{3} p_2 + \frac{1}{3} p_3\right) +$ 
    (autovectoresNormalizados[[1, 8]]) *  $\left(\frac{1}{3} p_3 + \frac{1}{3} p_3 + \frac{1}{3} p_3\right), \{\epsilon, 0, 1\}]$ 

Out[*]:=

$$\frac{\frac{a + a b - a c}{1 + 2 a - b + a b - c - 2 a c + b c} + (-1 - 2 a - a^2 - a b - a^2 b + b^2 + a b^2 + 2 c + 3 a c + a^2 c - b c - b^2 c - c^2 - a c^2 + b c^2) \epsilon}{(1 + 2 a - b + a b - c - 2 a c + b c)^2} + O[\epsilon]^2$$


```

Cálculo de la distribución estacionaria y la probabilidad de ganar del juego A+B'' en los juegos cooperativos

```

In[*]:= P = (1 / 2) * (A + B);

In[*]:= autovectores = NullSpace[Id - Transpose[P]];
|espacio nulo |transposición

In[*]:= autovectoresNormalizados = autovectores /
    (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
    autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]]);

```

```

In[ ]:= 1/2 * (1/2 - ε) + (1/2) * ((autovectoresNormalizados[[1, 1]]) * (1/3 p0 + 1/3 p0 + 1/3 p0) +
    (autovectoresNormalizados[[1, 2]]) * (1/3 p2 + 1/3 p1 + 1/3 p0) +
    (autovectoresNormalizados[[1, 3]]) * (1/3 p1 + 1/3 p0 + 1/3 p2) +
    (autovectoresNormalizados[[1, 4]]) * (1/3 p3 + 1/3 p1 + 1/3 p2) +
    (autovectoresNormalizados[[1, 5]]) * (1/3 p0 + 1/3 p2 + 1/3 p1) +
    (autovectoresNormalizados[[1, 6]]) * (1/3 p2 + 1/3 p3 + 1/3 p1) +
    (autovectoresNormalizados[[1, 7]]) * (1/3 p1 + 1/3 p2 + 1/3 p3) +
    (autovectoresNormalizados[[1, 8]]) * (1/3 p3 + 1/3 p3 + 1/3 p3));

```

```

In[ ]:= Simplify[
  simplifica

```

```

1/2 * (1/2 - ε) + (1/2) * ((autovectoresNormalizados[[1, 1]]) * (1/3 p0 + 1/3 p0 + 1/3 p0) +
    (autovectoresNormalizados[[1, 2]]) * (1/3 p2 + 1/3 p1 + 1/3 p0) +
    (autovectoresNormalizados[[1, 3]]) * (1/3 p1 + 1/3 p0 + 1/3 p2) +
    (autovectoresNormalizados[[1, 4]]) * (1/3 p3 + 1/3 p1 + 1/3 p2) +
    (autovectoresNormalizados[[1, 5]]) * (1/3 p0 + 1/3 p2 + 1/3 p1) +
    (autovectoresNormalizados[[1, 6]]) * (1/3 p2 + 1/3 p3 + 1/3 p1) +
    (autovectoresNormalizados[[1, 7]]) * (1/3 p1 + 1/3 p2 + 1/3 p3) +
    (autovectoresNormalizados[[1, 8]]) * (1/3 p3 + 1/3 p3 + 1/3 p3))];

```

```

Out[ ]:=

```

$$\frac{(2 + b - c) (1 + 2 a - 4 \epsilon)}{8 - 5 c + 2 b (-1 + c - 4 \epsilon) + 4 c \epsilon + a (7 + 2 b - 4 c + 4 \epsilon)}$$

```

In[ ]:= Series[
  serie

```

```

Out[ ]:=

```

$$\frac{(1 + 2 a) (2 + b - c)}{8 + a (7 + 2 b - 4 c) + 2 b (-1 + c) - 5 c} - \frac{8 ((2 + b - c) (4 + 4 a + a^2 - 2 b - a b - 2 c - a c + b c)) \epsilon}{(8 + 7 a - 2 b + 2 a b - 5 c - 4 a c + 2 b c)^2} + O[\epsilon]^2$$

¿ Hasta qué valor de ϵ se verifica la paradoja ?

`In[ ]:= NSolve[ $\frac{(2 + b - c) (1 + 2 a - 4 \epsilon)}{8 - 5 c + 2 b (-1 + c - 4 \epsilon) + 4 c \epsilon + a (7 + 2 b - 4 c + 4 \epsilon)}$  == 0.5,  $\epsilon$ ]`
[\[resuelve numéricamente\]](#)

`Out[ ]=`

$$\left\{ \left\{ \epsilon \rightarrow \frac{0.25 (-4. + a + 4. b + 2. a b + 3. c - 2. b c)}{4. + a - 1. c} \right\} \right\}$$

7.2.4. Condiciones para que el juego cooperativo sea justo

Estudiemos las condiciones que deben verificar las probabilidades de los juegos cooperativos en su primera variante para que se dé la paradoja de Parrondo

```
In[ ]:= p = 1 / 2 - ε;  
p0 = a - ε;  
p1 = b - ε;  
p2 = b - ε;  
p3 = c - ε;
```

```
In[ ]:= A =
```

$$\begin{pmatrix} 1-p & \frac{1}{3} * p & \frac{1}{3} * p & 0 & \frac{1}{3} \\ \frac{1}{3} * (1-p) & \frac{1}{3} * ((1-p) + (1-p) + p) & 0 & \frac{1}{3} * p & \\ \frac{1}{3} * (1-p) & 0 & \frac{1}{3} * ((1-p) + (1-p) + p) & \frac{1}{3} * p & \\ 0 & \frac{1}{3} * (1-p) & \frac{1}{3} * (1-p) & \frac{1}{3} * ((1-p) + p + p) & \\ \frac{1}{3} * (1-p) & 0 & 0 & 0 & \frac{1}{3} * ((1-p) \\ 0 & \frac{1}{3} * (1-p) & 0 & 0 & \frac{1}{3} * (\\ 0 & 0 & \frac{1}{3} * (1-p) & 0 & \frac{1}{3} * (\\ 0 & 0 & 0 & \frac{1}{3} * (1-p) & \end{pmatrix}$$

;

In[*]:= B =

$$\begin{pmatrix} 1 - p_0 & \frac{1}{3} * p_0 & \frac{1}{3} * p_0 & 0 & \frac{1}{3} * p_0 \\ \frac{1}{3} * (1 - p_0) & \frac{1}{3} * (2 - p_2 - p_1 + p_0) & 0 & \frac{1}{3} * p_1 & 0 \\ \frac{1}{3} * (1 - p_0) & 0 & \frac{1}{3} * (2 - p_2 - p_1 + p_0) & \frac{1}{3} * p_2 & 0 \\ 0 & \frac{1}{3} * (1 - p_1) & \frac{1}{3} * (1 - p_2) & \frac{1}{3} * (1 - p_3 + p_1 + p_2) & 0 \\ \frac{1}{3} * (1 - p_0) & 0 & 0 & 0 & \frac{1}{3} * (2 - p_2 - p_1 + p_0) \\ 0 & \frac{1}{3} * (1 - p_2) & 0 & 0 & \frac{1}{3} * (1 - p_1) \\ 0 & 0 & \frac{1}{3} * (1 - p_1) & 0 & \frac{1}{3} * (1 - p_2) \\ 0 & 0 & 0 & \frac{1}{3} * (1 - p_3) & 0 \end{pmatrix};$$

Sabemos que A es será justo o perdedor independientemente de a,b y c solamente dependerá de si ϵ es 0 o no. Calculemos la probabilidad de ganar en B

In[*]:= Id = IdentityMatrix[8];
matriz identidad

In[*]:= autovectores = NullSpace[Id - Transpose[B]];
espacio nulo transposición

In[*]:= autovectoresNormalizados = autovectores /
 (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
 autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]]);

```
In[*]:= probabilidadGanarB =  $\left(\frac{1}{3} p_0 + \frac{1}{3} p_0 + \frac{1}{3} p_0\right) * \text{autovectoresNormalizados}[[1, 1]] +$ 
```

```
 $\left(\frac{1}{3} p_2 + \frac{1}{3} p_1 + \frac{1}{3} p_0\right) * \text{autovectoresNormalizados}[[1, 2]] +$ 
```

```
 $\left(\frac{1}{3} p_1 + \frac{1}{3} p_0 + \frac{1}{3} p_2\right) * \text{autovectoresNormalizados}[[1, 3]] +$ 
```

```
 $\left(\frac{1}{3} p_3 + \frac{1}{3} p_1 + \frac{1}{3} p_2\right) * \text{autovectoresNormalizados}[[1, 4]] +$ 
```

```
 $\left(\frac{1}{3} p_0 + \frac{1}{3} p_2 + \frac{1}{3} p_1\right) * \text{autovectoresNormalizados}[[1, 5]] +$ 
```

```
 $\left(\frac{1}{3} p_2 + \frac{1}{3} p_3 + \frac{1}{3} p_1\right) * \text{autovectoresNormalizados}[[1, 6]] +$ 
```

```
 $\left(\frac{1}{3} p_1 + \frac{1}{3} p_2 + \frac{1}{3} p_3\right) * \text{autovectoresNormalizados}[[1, 7]] +$ 
```

```
 $\left(\frac{1}{3} p_3 + \frac{1}{3} p_3 + \frac{1}{3} p_3\right) * \text{autovectoresNormalizados}[[1, 8]];$ 
```

```
In[*]:= expresion1 = Series[probabilidadGanarB, {ε, 0, 1}]
```

```
Out[*]=
```

$$\frac{a + a b - a c}{1 + 2 a - b + a b - c - 2 a c + b c} + \frac{\left(-1 - 2 a - a^2 - a b - a^2 b + b^2 + a b^2 + 2 c + 3 a c + a^2 c - b c - b^2 c - c^2 - a c^2 + b c^2\right) \epsilon}{\left(1 + 2 a - b + a b - c - 2 a c + b c\right)^2} + O[\epsilon]^2$$

```
In[*]:= matrizAleatoria = (1/2) * A + (1/2) * B;
```

```
In[*]:= autovectores = NullSpace[Id - Transpose[matrizAleatoria]];
```

```
In[*]:= autovectoresNormalizados = autovectores /  
(autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +  
autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]]);
```

```
In[*]:= probabilidadGanarAleatoria =
```

```
 $(1/2) * (1/2 - \epsilon) + (1/2) * \left(\left(\frac{1}{3} p_0 + \frac{1}{3} p_0 + \frac{1}{3} p_0\right) * \text{autovectoresNormalizados}[[1, 1]] +$ 
```

```
 $\left(\frac{1}{3} p_2 + \frac{1}{3} p_1 + \frac{1}{3} p_0\right) * \text{autovectoresNormalizados}[[1, 2]] +$ 
```

```
 $\left(\frac{1}{3} p_1 + \frac{1}{3} p_0 + \frac{1}{3} p_2\right) * \text{autovectoresNormalizados}[[1, 3]] +$ 
```

```
 $\left(\frac{1}{3} p_3 + \frac{1}{3} p_1 + \frac{1}{3} p_2\right) * \text{autovectoresNormalizados}[[1, 4]] +$ 
```

```
 $\left(\frac{1}{3} p_0 + \frac{1}{3} p_2 + \frac{1}{3} p_1\right) * \text{autovectoresNormalizados}[[1, 5]] +$ 
```

```
 $\left(\frac{1}{3} p_2 + \frac{1}{3} p_3 + \frac{1}{3} p_1\right) * \text{autovectoresNormalizados}[[1, 6]] +$ 
```

```
 $\left(\frac{1}{3} p_1 + \frac{1}{3} p_2 + \frac{1}{3} p_3\right) * \text{autovectoresNormalizados}[[1, 7]] +$ 
```

```
 $\left(\frac{1}{3} p_3 + \frac{1}{3} p_3 + \frac{1}{3} p_3\right) * \text{autovectoresNormalizados}[[1, 8]]);$ 
```

```

In[ ]:= expresion2 = Series[probabilidadGanarAleatoria, {ϵ, 0, 1}];
[serie]

(* Damos el valor de epsilon 0 para ver cuando son justos *)

In[ ]:= ϵ = 0;

In[ ]:= expresion1
Out[ ]:=

$$\frac{a + a b - a c}{1 + 2 a - b + a b - c - 2 a c + b c}$$


In[ ]:= expresion2
Out[ ]:=

$$\frac{2 + 4 a + b + 2 a b - c - 2 a c}{8 + 7 a - 2 b + 2 a b - 5 c - 4 a c + 2 b c}$$


No hemos perdido precisión al trabajar con expresion1 y expresion2, pues aunque son desarrollos de Taylor los estamos evaluando en  $\epsilon = 0$ 

In[ ]:= Reduce[[ {(a + a b - a c) / (1 + 2 a - b + a b - c - 2 a c + b c) == 0.5,
[reduce]

$$\frac{2 + 4 a + b + 2 a b - c - 2 a c}{8 + 7 a - 2 b + 2 a b - 5 c - 4 a c + 2 b c} > 0.5, 0 \leq a \leq 1, 0 \leq b \leq 1, 0 \leq c \leq 1\}, \{a, b, c\}, \text{Reals}]$$

[números reales]

ⓘ Reduce: Reduce was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numerizing the result. ?

Out[ ]:=

$$c == \frac{1. - 1. b - 1. a b}{1. - 1. b}$$


```

Problemas que se tiene la paradoja de Parrondo con valores pertenecientes a estos intervalos, por ejemplo para $a = 1/4$, $b = 6/10$, $c = 5/8$

```

In[ ]:= a = 1 / 4;
        b = 6 / 10;
        c = 5 / 8;

In[ ]:= Clear[ϵ]
[borra]

In[ ]:= p = 1 / 2 - ϵ;
        p0 = a - ϵ;
        p1 = b - ϵ;
        p2 = b - ϵ;
        p3 = c - ϵ;

```

In[*]:= **A =**

$$\begin{pmatrix} 1-p & \frac{1}{3} * p & \frac{1}{3} * p & 0 & \frac{1}{3} \\ \frac{1}{3} * (1-p) & \frac{1}{3} * ((1-p) + (1-p) + p) & 0 & \frac{1}{3} * p & \\ \frac{1}{3} * (1-p) & 0 & \frac{1}{3} * ((1-p) + (1-p) + p) & \frac{1}{3} * p & \\ 0 & \frac{1}{3} * (1-p) & \frac{1}{3} * (1-p) & \frac{1}{3} * ((1-p) + p + p) & \\ \frac{1}{3} * (1-p) & 0 & 0 & 0 & \frac{1}{3} * ((1-p) \\ 0 & \frac{1}{3} * (1-p) & 0 & 0 & \frac{1}{3} * (\\ 0 & 0 & \frac{1}{3} * (1-p) & 0 & \frac{1}{3} * (\\ 0 & 0 & 0 & \frac{1}{3} * (1-p) & \end{pmatrix}$$

In[*]:= **B =**

$$\begin{pmatrix} 1-p0 & \frac{1}{3} * p0 & \frac{1}{3} * p0 & 0 & \frac{1}{3} * p0 \\ \frac{1}{3} * (1-p0) & \frac{1}{3} * (2-p2-p1+p0) & 0 & \frac{1}{3} * p1 & 0 \\ \frac{1}{3} * (1-p0) & 0 & \frac{1}{3} * (2-p2-p1+p0) & \frac{1}{3} * p2 & 0 \\ 0 & \frac{1}{3} * (1-p1) & \frac{1}{3} * (1-p2) & \frac{1}{3} * (1-p3+p1+p2) & 0 \\ \frac{1}{3} * (1-p0) & 0 & 0 & 0 & \frac{1}{3} * (2-p2-p1+p0) \\ 0 & \frac{1}{3} * (1-p2) & 0 & 0 & \frac{1}{3} * (1-p1) \\ 0 & 0 & \frac{1}{3} * (1-p1) & 0 & \frac{1}{3} * (1-p2) \\ 0 & 0 & 0 & \frac{1}{3} * (1-p3) & 0 \end{pmatrix}$$

In[*]:= **autovectores = NullSpace[Id - Transpose[B]];**
|espacio nulo |transposición

```
In[*]:= autovectoresNormalizados = autovectores /
  (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
   autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]]);
```

```
In[*]:= probabilidadGanarB =  $\left(\frac{1}{3} p_0 + \frac{1}{3} p_0 + \frac{1}{3} p_0\right) * \text{autovectoresNormalizados}[[1, 1]] +$ 
 $\left(\frac{1}{3} p_2 + \frac{1}{3} p_1 + \frac{1}{3} p_0\right) * \text{autovectoresNormalizados}[[1, 2]] +$ 
 $\left(\frac{1}{3} p_1 + \frac{1}{3} p_0 + \frac{1}{3} p_2\right) * \text{autovectoresNormalizados}[[1, 3]] +$ 
 $\left(\frac{1}{3} p_3 + \frac{1}{3} p_1 + \frac{1}{3} p_2\right) * \text{autovectoresNormalizados}[[1, 4]] +$ 
 $\left(\frac{1}{3} p_0 + \frac{1}{3} p_2 + \frac{1}{3} p_1\right) * \text{autovectoresNormalizados}[[1, 5]] +$ 
 $\left(\frac{1}{3} p_2 + \frac{1}{3} p_3 + \frac{1}{3} p_1\right) * \text{autovectoresNormalizados}[[1, 6]] +$ 
 $\left(\frac{1}{3} p_1 + \frac{1}{3} p_2 + \frac{1}{3} p_3\right) * \text{autovectoresNormalizados}[[1, 7]] +$ 
 $\left(\frac{1}{3} p_3 + \frac{1}{3} p_3 + \frac{1}{3} p_3\right) * \text{autovectoresNormalizados}[[1, 8]]$ 
```

```
Out[*]= 
$$\frac{\frac{5}{8} - \epsilon}{1 - \frac{3(3+8\epsilon)}{-5+8\epsilon} - \frac{3(-6-31\epsilon-40\epsilon^2)}{(-3+5\epsilon)(-5+8\epsilon)} - \frac{18+117\epsilon+244\epsilon^2+160\epsilon^3}{(-1+4\epsilon)(-3+5\epsilon)(-5+8\epsilon)}} -$$


$$\frac{3\left(\frac{2}{3}\left(\frac{3}{5} - \epsilon\right) + \frac{1}{3}\left(\frac{5}{8} - \epsilon\right)\right)(3+8\epsilon)}{(-5+8\epsilon)\left(1 - \frac{3(3+8\epsilon)}{-5+8\epsilon} - \frac{3(-6-31\epsilon-40\epsilon^2)}{(-3+5\epsilon)(-5+8\epsilon)} - \frac{18+117\epsilon+244\epsilon^2+160\epsilon^3}{(-1+4\epsilon)(-3+5\epsilon)(-5+8\epsilon)}\right)} -$$


$$\frac{3\left(\frac{1}{3}\left(\frac{1}{4} - \epsilon\right) + \frac{2}{3}\left(\frac{3}{5} - \epsilon\right)\right)(-6-31\epsilon-40\epsilon^2)}{(-3+5\epsilon)(-5+8\epsilon)\left(1 - \frac{3(3+8\epsilon)}{-5+8\epsilon} - \frac{3(-6-31\epsilon-40\epsilon^2)}{(-3+5\epsilon)(-5+8\epsilon)} - \frac{18+117\epsilon+244\epsilon^2+160\epsilon^3}{(-1+4\epsilon)(-3+5\epsilon)(-5+8\epsilon)}\right)} -$$


$$\frac{\left(\frac{1}{4} - \epsilon\right)(18+117\epsilon+244\epsilon^2+160\epsilon^3)}{(-1+4\epsilon)(-3+5\epsilon)(-5+8\epsilon)\left(1 - \frac{3(3+8\epsilon)}{-5+8\epsilon} - \frac{3(-6-31\epsilon-40\epsilon^2)}{(-3+5\epsilon)(-5+8\epsilon)} - \frac{18+117\epsilon+244\epsilon^2+160\epsilon^3}{(-1+4\epsilon)(-3+5\epsilon)(-5+8\epsilon)}\right)}$$

```

```
In[*]:= expresion1 = Series[probabilidadGanarB, {ϵ, 0, 1}]
[serie]
```

```
Out[*]= 
$$\frac{1}{2} - \frac{5\epsilon}{3} + O[\epsilon]^2$$

```

Efectivamente el juego B es justo

```
In[*]:= matrizAleatoria = (1/2) * A + (1/2) * B;
```

```
In[*]:= Id = IdentityMatrix[8];
[matriz identidad]
```

```
In[*]:= autovectores = NullSpace[Id - Transpose[matrizAleatoria]];
[espacio nulo] [transposición]
```

```
In[*]:= autovectoresNormalizados = autovectores /
  (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
    autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]]);
```

```
In[*]:= probabilidadGanarAleatoria =
  (1 / 2) * (1 / 2 - ε) + (1 / 2) *  $\left( \left( \frac{1}{3} p_0 + \frac{1}{3} p_0 + \frac{1}{3} p_0 \right) * \text{autovectoresNormalizados}[[1, 1]] + \right.$ 
 $\left( \frac{1}{3} p_2 + \frac{1}{3} p_1 + \frac{1}{3} p_0 \right) * \text{autovectoresNormalizados}[[1, 2]] +$ 
 $\left( \frac{1}{3} p_1 + \frac{1}{3} p_0 + \frac{1}{3} p_2 \right) * \text{autovectoresNormalizados}[[1, 3]] +$ 
 $\left( \frac{1}{3} p_3 + \frac{1}{3} p_1 + \frac{1}{3} p_2 \right) * \text{autovectoresNormalizados}[[1, 4]] +$ 
 $\left( \frac{1}{3} p_0 + \frac{1}{3} p_2 + \frac{1}{3} p_1 \right) * \text{autovectoresNormalizados}[[1, 5]] +$ 
 $\left( \frac{1}{3} p_2 + \frac{1}{3} p_3 + \frac{1}{3} p_1 \right) * \text{autovectoresNormalizados}[[1, 6]] +$ 
 $\left( \frac{1}{3} p_1 + \frac{1}{3} p_2 + \frac{1}{3} p_3 \right) * \text{autovectoresNormalizados}[[1, 7]] +$ 
 $\left. \left( \frac{1}{3} p_3 + \frac{1}{3} p_3 + \frac{1}{3} p_3 \right) * \text{autovectoresNormalizados}[[1, 8]] \right)$ 
```

```
Out[*]=  $\frac{1}{2} \left( \frac{1}{2} - \epsilon \right) + \frac{1}{2} \left( \frac{\left( \frac{1}{4} - \epsilon \right) \left( -315 - 1924 \epsilon - 3872 \epsilon^2 - 2560 \epsilon^3 \right)}{-1872 + 416 \epsilon} + \right.$ 
 $\frac{3 \left( \frac{2}{3} \left( \frac{3}{5} - \epsilon \right) + \frac{1}{3} \left( \frac{5}{8} - \epsilon \right) \right) \left( -231 + 508 \epsilon + 1248 \epsilon^2 - 2560 \epsilon^3 \right)}{-1872 + 416 \epsilon} +$ 
 $\frac{\left( \frac{5}{8} - \epsilon \right) \left( -297 + 1860 \epsilon - 3808 \epsilon^2 + 2560 \epsilon^3 \right)}{-1872 + 416 \epsilon} +$ 
 $\left. \frac{3 \left( \frac{1}{3} \left( \frac{1}{4} - \epsilon \right) + \frac{2}{3} \left( \frac{3}{5} - \epsilon \right) \right) \left( -189 - 348 \epsilon + 1312 \epsilon^2 + 2560 \epsilon^3 \right)}{-1872 + 416 \epsilon} \right)$ 
```

```
In[*]:= expresion2 = Series[probabilidadGanarAleatoria, {ε, 0, 1}]
serie
```

```
Out[*]=  $\frac{79}{156} - \frac{869 \epsilon}{702} + 0[\epsilon]^2$ 
```

```
In[*]:= N[Series[probabilidadGanarAleatoria, {ε, 0, 1}]]
serie
```

```
Out[*]= 0.50641 - 1.23789 (ε + 0.) + 0[ε + 0.]2
```

Efectivamente la mezcla aleatoria es un juego ganador para valores de ϵ pequeños

¿ Hasta qué valor de ϵ se verifica la paradoja ?

```
In[ ]:= Solve[probabilidadGanarAleatoria == 0.5,  $\epsilon$ ]
```

```
|resuelve
```

 **Solve:** Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

```
Out[ ]:=
```

```
{ { $\epsilon \rightarrow 0.00517241$ } }
```

7.2.5. Distribuciones estacionarias en los juegos cooperativos en su segunda variante

Calculamos la distribución estacionaria del juego A en la segunda variante del caso cooperativo

$$\text{In[*]:= } \mathbf{A} = \begin{pmatrix} 0 & 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 1/3 & 1/6 & 1/6 & 1/6 & 1/6 & 0 & 0 \\ 0 & 1/6 & 1/3 & 1/6 & 1/6 & 0 & 1/6 & 0 \\ 0 & 1/6 & 1/6 & 1/3 & 0 & 1/6 & 1/6 & 0 \\ 0 & 1/6 & 1/6 & 0 & 1/3 & 1/6 & 1/6 & 0 \\ 0 & 1/6 & 0 & 1/6 & 1/6 & 1/3 & 1/6 & 0 \\ 0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 1/3 & 1/3 & 0 \end{pmatrix};$$

`In[*]:= Id = IdentityMatrix[8];`
[matriz identidad]

`In[*]:= autovectores = NullSpace[Id - Transpose[A]];`
[espacio nulo] [transposición]

`In[*]:= autovectoresNormalizados = autovectores /`
`(autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +`
`autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]])`

`Out[*]=`
 $\left\{ \left\{ 0, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, 0 \right\} \right\}$

Observamos que efectivamente la distribución estacionaria es $(0, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 0)$

7.2.6. Estudio del parámetro γ en los juegos originales

Cálculo de la probabilidad de ganar en la combinación aleatoria de A y B, sin determinar γ , con $\epsilon = 0$

```
In[1]:=  $\epsilon = 0$ ;

In[2]:=  $pa = 0.5 - \epsilon$ ;
 $pb = 0.75 - \epsilon$ ;
 $pm = 0.1 - \epsilon$ ;

In[5]:=  $A = \begin{pmatrix} 0 & pa & 1 - pa \\ 1 - pa & 0 & pa \\ pa & 1 - pa & 0 \end{pmatrix}$ ;

In[6]:=  $B = \begin{pmatrix} 0 & pm & 1 - pm \\ 1 - pb & 0 & pb \\ pb & 1 - pb & 0 \end{pmatrix}$ ;

In[7]:=  $matrizAleatoria = (\gamma) * (A) + (1 - \gamma) * B$ ;

In[8]:=  $Id = IdentityMatrix[3]$ ;
 $\quad \quad \quad \text{[matriz identidad]}$ 

In[9]:=  $autovectores = NullSpace[Id - Transpose[matrizAleatoria]]$ ;
 $\quad \quad \quad \text{[espacio nulo] [transposición]}$ 

In[10]:=  $autovectoresNormalizados =$ 
 $\quad \quad \quad autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])$ 

Out[10]= 
$$\left\{ \left\{ -\frac{0.625 (13. - 2. \gamma + 1. \gamma^2)}{(-9.75 + 1.25 \gamma + 1. \gamma^2) \left( 1. + \frac{1. (-3.25 - 5.25 \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} - \frac{0.625 (13. - 2. \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} \right)}, \right.$$


$$\frac{1. (-3.25 - 5.25 \gamma + 1. \gamma^2)}{(-9.75 + 1.25 \gamma + 1. \gamma^2) \left( 1. + \frac{1. (-3.25 - 5.25 \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} - \frac{0.625 (13. - 2. \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} \right)},$$


$$\left. \frac{1.}{1. + \frac{1. (-3.25 - 5.25 \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} - \frac{0.625 (13. - 2. \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2}} \right\}$$


In[11]:=  $(\gamma) (1/2 - \epsilon) + (1 - \gamma) * ((autovectoresNormalizados[[1, 1]]) * (1/10 - \epsilon) +$ 
 $\quad \quad \quad (1 - (autovectoresNormalizados[[1, 1]])) * ((3/4) - \epsilon))$ 

Out[11]= 
$$\frac{\gamma}{2} + (1 - \gamma) \left( -\frac{0.0625 (13. - 2. \gamma + 1. \gamma^2)}{(-9.75 + 1.25 \gamma + 1. \gamma^2) \left( 1. + \frac{1. (-3.25 - 5.25 \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} - \frac{0.625 (13. - 2. \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} \right)} + \right.$$


$$\left. \frac{3}{4} \left( 1 + \frac{0.625 (13. - 2. \gamma + 1. \gamma^2)}{(-9.75 + 1.25 \gamma + 1. \gamma^2) \left( 1. + \frac{1. (-3.25 - 5.25 \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} - \frac{0.625 (13. - 2. \gamma + 1. \gamma^2)}{-9.75 + 1.25 \gamma + 1. \gamma^2} \right)} \right) \right)$$

```

Cálculo de la probabilidad de ganar en la combinación

aleatoria de A y B, sin determinar γ ni ϵ

```

In[12]:= Clear[ $\epsilon$ ]
borra

In[13]:= pa = 0.5 -  $\epsilon$ ;
pb = 0.75 -  $\epsilon$ ;
pm = 0.1 -  $\epsilon$ ;

In[16]:= A =  $\begin{pmatrix} 0 & pa & 1 - pa \\ 1 - pa & 0 & pa \\ pa & 1 - pa & 0 \end{pmatrix}$ ;

In[17]:= B =  $\begin{pmatrix} 0 & pm & 1 - pm \\ 1 - pb & 0 & pb \\ pb & 1 - pb & 0 \end{pmatrix}$ ;

In[18]:= matrizAleatoria = ( $\gamma$ ) * (A) + (1 -  $\gamma$ ) * B;

In[19]:= autovectores = NullSpace[Id - Transpose[matrizAleatoria]];
espacio nulo transposición

In[20]:= autovectoresNormalizados =
autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])

Out[20]=

$$\left\{ \left\{ \left( 0.0625 \left( 13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2 \right) \right) / \right. \right.$$


$$\left( 0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2 \right)$$


$$\left( 1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right.$$


$$\left. \frac{0.0625 \left( 13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2 \right)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \Bigg),$$


$$- \left( (-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2) / \right.$$


$$\left( 0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2 \right)$$


$$\left( 1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right.$$


$$\left. \frac{0.0625 \left( 13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2 \right)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \Bigg) \Bigg),$$


$$\frac{1.}{1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \frac{0.0625 \left( 13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2 \right)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2}} \Bigg\}$$


```

In[21]:= **Series**[**autovectoresNormalizados**[[1, 1]], { ϵ , 0, 2}]
|serie

$$\text{Out[21]} = \frac{0.454545 (13. - 2. \gamma + 1. \gamma^2)}{-15.3636 - 2. \gamma + 1. \gamma^2} - \frac{4.29752 (-107.25 + 101.5 \gamma + 29. \gamma^2 - 22.5 \gamma^3 - 1.75 \gamma^4 + 1. \gamma^5) \epsilon}{(-15.3636 - 2. \gamma + 1. \gamma^2)^2 (-9.75 + 1.25 \gamma + 1. \gamma^2)} - \frac{(23.441 (-1349.89 + 2855.77 \gamma - 1263.36 \gamma^2 - 665.05 \gamma^3 + 409.388 \gamma^4 + 50.775 \gamma^5 - 37.1375 \gamma^6 - 1.5 \gamma^7 + 1. \gamma^8) \epsilon^2)}{((-15.3636 - 2. \gamma + 1. \gamma^2)^3 (-9.75 + 1.25 \gamma + 1. \gamma^2)^2) + 0[\epsilon]^3}$$

In[22]:= **Series**[**autovectoresNormalizados**[[1, 2]], { ϵ , 0, 2}]
|serie

$$\text{Out[22]} = \frac{0.727273 (-3.25 - 5.25 \gamma + 1. \gamma^2)}{-15.3636 - 2. \gamma + 1. \gamma^2} + \frac{2.14876 (-91.65 + 68.3 \gamma + 47. \gamma^2 - 21.3 \gamma^3 - 3.35 \gamma^4 + 1. \gamma^5) \epsilon}{(-15.3636 - 2. \gamma + 1. \gamma^2)^2 (-9.75 + 1.25 \gamma + 1. \gamma^2)} + \frac{(11.7205 (-7735.55 + 9961.12 \gamma - 83.9858 \gamma^2 - 2676.53 \gamma^3 + 343.929 \gamma^4 + 231.073 \gamma^5 - 34.7242 \gamma^6 - 6.32667 \gamma^7 + 1. \gamma^8) \epsilon^2)}{((-15.3636 - 2. \gamma + 1. \gamma^2)^3 (-9.75 + 1.25 \gamma + 1. \gamma^2)^2) + 0[\epsilon]^3}$$

In[23]:= **Series**[**autovectoresNormalizados**[[1, 3]], { ϵ , 0, 2}]
|serie

$$\text{Out[23]} = \frac{0.727273 (-9.75 + 1.25 \gamma + 1. \gamma^2)}{-15.3636 - 2. \gamma + 1. \gamma^2} + \frac{2.14876 (12.6 - 12.2 \gamma - 1.4 \gamma^2 + 1. \gamma^3) \epsilon}{(-15.3636 - 2. \gamma + 1. \gamma^2)^2} + \frac{11.7205 (-516.49 + 369.637 \gamma + 244.953 \gamma^2 - 68.78 \gamma^3 - 32.3967 \gamma^4 + 2.07667 \gamma^5 + 1. \gamma^6) \epsilon^2}{((-15.3636 - 2. \gamma + 1. \gamma^2)^3 (-9.75 + 1.25 \gamma + 1. \gamma^2)^2) + 0[\epsilon]^3}$$

Calculemos la probabilidad total de ganar en el juego aleatorizado

```
In[24]:= (γ) * (1 / 2 - ε) + (1 - γ) * ((autovectoresNormalizados[[1, 1]]) * (1 / 10 - ε) +
(1 - (autovectoresNormalizados[[1, 1]])) * ((3 / 4) - ε))
```

```
Out[24]=
```

$$\begin{aligned} & \gamma \left(\frac{1}{2} - \epsilon \right) + (1 - \gamma) \left(\left(0.0625 \left(\frac{1}{10} - \epsilon \right) (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2) \right) / \left((0.975 - 0.125 \gamma - 0.1 \gamma^2 + \right. \right. \\ & \quad \left. \left. 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2) \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \right. \right. \\ & \quad \left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) + \\ & \quad \left(\frac{3}{4} - \epsilon \right) \left(1 - (0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)) / \right. \\ & \quad \left((0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2) \right. \\ & \quad \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \\ & \quad \left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) \right) \end{aligned}$$

La siguiente celda hay que darla de forma explícita o si no no lo reconoce. En esta primera celda podemos ver el resultado de la probabilidad según se van variando γ y ϵ , y en la segunda tenemos el gráfico

In[25]:= **Manipulate**[
[manipula](#)

$$\gamma \left(\frac{1}{2} - \epsilon \right) + (1 - \gamma) \left(\left(0.0625 \left(\frac{1}{10} - \epsilon \right) (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2) \right) / \right. \\
\left. \left((0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2) \right. \right. \\
\left. \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \right. \\
\left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) + \\
\left(\frac{3}{4} - \epsilon \right) \left(1 - (0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)) / \right. \\
\left((0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2) \right. \\
\left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \\
\left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) \right), \\
\{\epsilon, 0, 0.1\}, \{\gamma, 0, 1\}]$$

Out[25]=

The image shows a Mathematica Manipulate interface. It has two horizontal sliders: the top one is labeled with the Greek letter epsilon (ε) and the bottom one with the Greek letter gamma (γ). Below the sliders is a rectangular display box containing the number 0.5. In the top right corner of the interface, there is a small circular button with a plus sign (+).

In[29]:= **Manipulate****[****Plot****[**
[manipula [representación gráfica

$$\gamma \left(\frac{1}{2} - \epsilon \right) + (1 - \gamma) \left(\left(0.0625 \left(\frac{1}{10} - \epsilon \right) (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2) \right) / \right. \\ \left. \left(0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2 \right) \right. \\ \left. \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \right. \\ \left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) + \\ \left(\frac{3}{4} - \epsilon \right) \left(1 - (0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)) / \right. \\ \left. \left(0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2 \right) \right. \\ \left. \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \right. \\ \left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) \right), \\ \{\gamma, 0, 1\}, \{\epsilon, 0, 0.1\}];$$

¿Donde se da y cuál es el máximo de dicha probabilidad, en función de ϵ ?

Calculemos donde se da el máximo en función de ϵ empleando el comando Manipulate

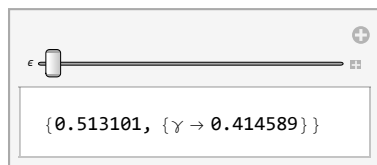
In[27]:= **probabilidad**[γ _, ϵ _] :=

$$\gamma \left(\frac{1}{2} - \epsilon \right) + (1 - \gamma) \left(\left(0.0625 \left(\frac{1}{10} - \epsilon \right) (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2) \right) / \right. \\ \left. \left((0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2) \right. \right. \\ \left. \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \right. \\ \left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) + \\ \left(\frac{3}{4} - \epsilon \right) \left(1 - (0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)) / \right. \\ \left. \left((0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2) \right. \right. \\ \left. \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \right. \\ \left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) \right);$$

In[28]:=

```
Manipulate[Module[{maximo}, maximo = FindMaximum[{probabilidad[ $\gamma$ ,  $\epsilon$ ], 0 ≤  $\gamma$  ≤ 1},  $\gamma$ ];
manipula      módulo      encuentra máximo
{maximo[[1]], maximo[[2]]}, {{ $\epsilon$ , 0, "ε"}, 0, 0.1, 0.001}]
```

Out[28]=



Representación en el plano de coordenadas ϵ y γ de los puntos donde esta probabilidad es mayor o igual que 0.5

```
In[32]:= RegionPlot[( $\gamma$ ) * (1 / 2 -  $\epsilon$ ) + (1 -  $\gamma$ ) * ((autovectoresNormalizados[[1, 1]]) * (1 / 10 -  $\epsilon$ ) +
representación gráfica de una región
(1 - (autovectoresNormalizados[[1, 1]])) * ((3 / 4) -  $\epsilon$ )) ≥ 0.5,
{ $\gamma$ , 0, 1}, {{ $\epsilon$ , 0, 0.015}, PlotPoints → 50, Mesh → None, AxesLabel → {" $\gamma$ ", "ε"},
número de puntos en malla ning... Etiqueta de ejes
PlotStyle → Directive[Opacity[0.7], Blue],
estilo de repr... directiva Opacidad azul
Frame → True, FrameLabel → {" $\gamma$ ", "ε"}];
marco verd... Etiqueta de marco
```

Estudiemos en qué valor de γ se da el mayor rango de valores posibles de ϵ y cual es ese valor máximo de ϵ

```

In[ ]:= ContourPlot[
  representación de contornos

  
$$\gamma \left( \frac{1}{2} - \epsilon \right) + (1 - \gamma) \left( \left( 0.0625 \left( \frac{1}{10} - \epsilon \right) (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2) \right) / \right.$$


$$\left( 0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2 \right)$$


$$\left( 1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right.$$


$$\left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) +$$


$$\left( \frac{3}{4} - \epsilon \right) \left( 1 - (0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)) / \right.$$


$$\left( 0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2 \right)$$


$$\left( 1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right.$$


$$\left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) \right) -$$

  0.5 == 0, {γ, 0, 1}, {ε, 0, 0.015}, ContourStyle → Directive[
    estilo de contorno  directiva
    Blue,
    azul
    Thick],
    grueso
  FrameLabel → {"γ", "ε"}, PlotPoints →
    etiqueta de marco  número de puntos en la representación
  50, Axes →
    ejes
  True];
verdadero

```

In[31]:= **FindMaximum**[
 encuentra máximo

$$\left\{ \epsilon, \gamma \left(\frac{1}{2} - \epsilon \right) + (1 - \gamma) \left(\left(0.0625 \left(\frac{1}{10} - \epsilon \right) (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2) \right) / \right. \right. \\ \left. \left(0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2 \right) \right. \\ \left. \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \right. \\ \left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) + \\ \left(\frac{3}{4} - \epsilon \right) \left(1 - (0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)) / \right. \\ \left. \left(0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2 \right) \right. \\ \left. \left(1. - \frac{-0.325 - 0.525 \gamma + 0.1 \gamma^2 - 0.15 \epsilon + 0.15 \gamma \epsilon - 1. \epsilon^2}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} + \right. \right. \\ \left. \left. \frac{0.0625 (13. - 2. \gamma + 1. \gamma^2 - 8. \epsilon + 8. \gamma \epsilon + 16. \epsilon^2)}{0.975 - 0.125 \gamma - 0.1 \gamma^2 + 0.15 \epsilon - 0.15 \gamma \epsilon + 1. \epsilon^2} \right) \right) \right) == \\ 0.5, 0 \leq \gamma \leq 1, 0 \leq \epsilon \leq 0.02 \}, \{ \{ \gamma, 0.5 \}, \{ \epsilon, 0.003 \} \}]$$

Out[31]=

{0.0137176, { $\gamma \rightarrow 0.396737$, $\epsilon \rightarrow 0.0137176$ }}

7.2.7. Estudio del parámetro γ en los juegos dependientes de la historia

—bigskip

Cálculo de la probabilidad de ganar en la combinación aleatoria de A y B', sin determinar γ ni ϵ

```
In[1]:= p = 1 / 2 -  $\epsilon$ ;

In[2]:= A =  $\begin{pmatrix} 1-p & p & 0 & 0 \\ 0 & 0 & 1-p & p \\ 1-p & p & 0 & 0 \\ 0 & 0 & 1-p & p \end{pmatrix}$ ;

In[3]:= p1 = 9 / 10 -  $\epsilon$ ;
p2 = 1 / 4 -  $\epsilon$ ;
p3 = 1 / 4 -  $\epsilon$ ;
p4 = 7 / 10 -  $\epsilon$ ;

In[7]:= B =  $\begin{pmatrix} 1-p1 & p1 & 0 & 0 \\ 0 & 0 & 1-p2 & p2 \\ 1-p3 & p3 & 0 & 0 \\ 0 & 0 & 1-p4 & p4 \end{pmatrix}$ ;

In[8]:= matrizAleatoria = ( $\gamma$ ) * (A) + (1 -  $\gamma$ ) * B;

In[9]:= Id = IdentityMatrix[4];
      |matriz identidad

In[10]:= autovectores = NullSpace[Id - Transpose[matrizAleatoria]];
      |espacio nulo      |transposición

In[11]:= autovectoresNormalizados = autovectores /
      (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]])

Out[11]= 
$$\left\{ \left\{ \frac{(-3 + \gamma - 4\epsilon)(3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon) \left( 1 + \frac{4(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon)} + \frac{(-3 + \gamma - 4\epsilon)(3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon)} \right)}, \right. \right.$$


$$\frac{2(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon) \left( 1 + \frac{4(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon)} + \frac{(-3 + \gamma - 4\epsilon)(3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon)} \right)},$$


$$\frac{2(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon) \left( 1 + \frac{4(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon)} + \frac{(-3 + \gamma - 4\epsilon)(3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon)} \right)}, \frac{1}{1 + \frac{4(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon)} + \frac{(-3 + \gamma - 4\epsilon)(3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon)}} \left. \right\}$$


In[12]:= ( $\gamma$ ) * (1 / 2 -  $\epsilon$ ) + (1 -  $\gamma$ ) * ((autovectoresNormalizados[[1, 1]]) * (p1) +
      (autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]]) * (p2) +
      (autovectoresNormalizados[[1, 4]]) * p4)

Out[12]= 
$$\gamma \left( \frac{1}{2} - \epsilon \right) + (1 - \gamma)$$


$$\left( \frac{\frac{7}{10} - \epsilon}{1 + \frac{4(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon)} + \frac{(-3 + \gamma - 4\epsilon)(3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon)}} + \frac{4 \left( \frac{1}{4} - \epsilon \right) (3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon) \left( 1 + \frac{4(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon)} + \frac{(-3 + \gamma - 4\epsilon)(3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon)} \right)} + \right.$$


$$\left. \frac{(-3 + \gamma - 4\epsilon) \left( \frac{9}{10} - \epsilon \right) (3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon) \left( 1 + \frac{4(3 + 2\gamma + 10\epsilon)}{5(1 + \gamma - 4\epsilon)} + \frac{(-3 + \gamma - 4\epsilon)(3 + 2\gamma + 10\epsilon)}{(1 + \gamma - 4\epsilon)(-9 + 4\gamma + 10\epsilon)} \right)} \right)$$

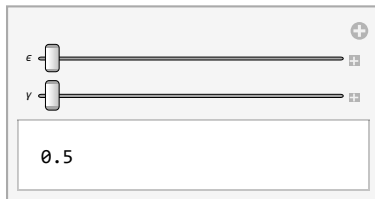
```

In[13]:= Manipulate[$\gamma \left(\frac{1}{2} - \epsilon \right) + (1 - \gamma)$
manipula

$$\left(\frac{\frac{7}{10} - \epsilon}{1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}} + \frac{4\left(\frac{1}{4} - \epsilon\right)(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)\left(1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}\right)} + \frac{(-3+\gamma-4\epsilon)\left(\frac{9}{10} - \epsilon\right)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)\left(1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}\right)} \right),$$

{ ϵ , 0, 0.25}, { γ , 0, 1}

Out[13]=



In[14]:= Manipulate[Plot[$\gamma \left(\frac{1}{2} - \epsilon \right) + (1 - \gamma)$
manipula representación grafica

$$\left(\frac{\frac{7}{10} - \epsilon}{1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}} + \frac{4\left(\frac{1}{4} - \epsilon\right)(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)\left(1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}\right)} + \frac{(-3+\gamma-4\epsilon)\left(\frac{9}{10} - \epsilon\right)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)\left(1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}\right)} \right),$$

{ γ , 0, 1}, { ϵ , 0, 0.25}];

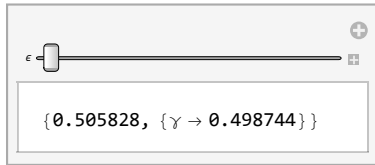
¿Dónde se da y cuál es el máximo de dicha probabilidad, en función de ϵ ?

In[15]:= probabilidad[γ _, ϵ _] := $\gamma \left(\frac{1}{2} - \epsilon \right) + (1 - \gamma)$

$$\left(\frac{\frac{7}{10} - \epsilon}{1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}} + \frac{4\left(\frac{1}{4} - \epsilon\right)(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)\left(1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}\right)} + \frac{(-3+\gamma-4\epsilon)\left(\frac{9}{10} - \epsilon\right)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)\left(1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}\right)} \right);$$

```
In[16]:= Manipulate[Module[{maximo}, maximo = FindMaximum[{probabilidad[γ, ε], 0 ≤ γ ≤ 1}, γ];
  {maximo[[1]], maximo[[2]]}], {{ε, 0, "ε"}, 0, 0.25, 0.001}]
```

Out[16]=



Representación en el plano de coordenadas ϵ y γ de los puntos donde esta probabilidad es mayor o igual que 0.5

```
In[23]:= RegionPlot[γ (1/2 - ε) + (1 - γ) (
  1 + 4 (3+2 γ+10 ε) / (5 (1+γ-4 ε)) + (-3+γ-4 ε) (3+2 γ+10 ε) / ((1+γ-4 ε) (-9+4 γ+10 ε))
) ≥ 0.5,
  {γ, 0, 1}, {ε, 0, 0.007}, PlotPoints → 50, Mesh → None,
  AxesLabel → {"γ", "ε"},
  PlotStyle → Directive[Opacity[0.7], Blue],
  Frame → True, FrameLabel → {"γ", "ε"}];
```

Estudiemos en qué valor de γ se da el mayor rango de valores posibles de ϵ y cual es ese valor máximo de ϵ

```
In[20]:= ContourPlot[ $\gamma \left( \frac{1}{2} - \epsilon \right) + (1 - \gamma) \left( \frac{\frac{7}{10} - \epsilon}{1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}} + \right.$ 
 $\frac{4 \left( \frac{1}{4} - \epsilon \right) (3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon) \left( 1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)} \right)} +$ 
 $\left. \frac{(-3+\gamma-4\epsilon) \left( \frac{9}{10} - \epsilon \right) (3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon) \left( 1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)} \right)} \right) - 0.5 == 0,$ 
 $\{\gamma, 0, 1\}, \{\epsilon, 0, 0.007\}, \text{ContourStyle} \rightarrow \text{Directive}[\text{Blue}, \text{Thick}],$ 
 $\text{FrameLabel} \rightarrow \{\gamma, \epsilon\},$ 
 $\text{PlotPoints} \rightarrow 50, \text{Axes} \rightarrow \text{True}];$ 
```

```
In[21]:= FindMaximum[ $\{\epsilon, \gamma \left( \frac{1}{2} - \epsilon \right) + (1 - \gamma) \left( \frac{\frac{7}{10} - \epsilon}{1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)}} + \right.$ 
 $\frac{4 \left( \frac{1}{4} - \epsilon \right) (3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon) \left( 1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)} \right)} +$ 
 $\left. \frac{(-3+\gamma-4\epsilon) \left( \frac{9}{10} - \epsilon \right) (3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon) \left( 1 + \frac{4(3+2\gamma+10\epsilon)}{5(1+\gamma-4\epsilon)} + \frac{(-3+\gamma-4\epsilon)(3+2\gamma+10\epsilon)}{(1+\gamma-4\epsilon)(-9+4\gamma+10\epsilon)} \right)} \right) == 0.5,$ 
 $0 \leq \gamma \leq 1, 0 \leq \epsilon \leq 0.007\}, \{\gamma, 0.5\}, \{\epsilon, 0.003\}]$ 
```

```
Out[21]= {0.00595576, { $\gamma \rightarrow 0.511911, \epsilon \rightarrow 0.00595576$ }}
```

7.2.8. Estudio del parámetro γ en los juegos cooperativos

Cálculo de la probabilidad de ganar en la combinación aleatoria de A y B'', en los juegos cooperativos en su primera variante, sin determinar γ ni ϵ , con los parámetros $a = 1/4$, $b = 6/10$, $c = 5/8$

In[1]:= $p = 1/2 - \epsilon$

Out[1]= $\frac{1}{2} - \epsilon$

In[2]:= $A =$

$$\begin{pmatrix} 1-p & \frac{1}{3} * p & \frac{1}{3} * p & 0 & \frac{1}{3} \\ \frac{1}{3} * (1-p) & \frac{1}{3} * ((1-p) + (1-p) + p) & 0 & \frac{1}{3} * p & \\ \frac{1}{3} * (1-p) & 0 & \frac{1}{3} * ((1-p) + (1-p) + p) & \frac{1}{3} * p & \\ 0 & \frac{1}{3} * (1-p) & \frac{1}{3} * (1-p) & \frac{1}{3} * ((1-p) + p + p) & \frac{1}{3} * ((1-p)) \\ \frac{1}{3} * (1-p) & 0 & 0 & 0 & \frac{1}{3} * ((1-p)) \\ 0 & \frac{1}{3} * (1-p) & 0 & 0 & \frac{1}{3} * ((1-p)) \\ 0 & 0 & \frac{1}{3} * (1-p) & 0 & \frac{1}{3} * ((1-p)) \\ 0 & 0 & 0 & \frac{1}{3} * (1-p) & \end{pmatrix};$$

In[3]:= $p0 = 1/4 - \epsilon;$

$p1 = 6/10 - \epsilon;$

$p2 = 6/10 - \epsilon;$

$p3 = 5/8 - \epsilon;$

In[7]:= B =

$$\begin{pmatrix} 1 - p_0 & \frac{1}{3} * p_0 & \frac{1}{3} * p_0 & 0 & \frac{1}{3} * p_0 \\ \frac{1}{3} * (1 - p_0) & \frac{1}{3} * (2 - p_2 - p_1 + p_0) & 0 & \frac{1}{3} * p_1 & 0 \\ \frac{1}{3} * (1 - p_0) & 0 & \frac{1}{3} * (2 - p_2 - p_1 + p_0) & \frac{1}{3} * p_2 & 0 \\ 0 & \frac{1}{3} * (1 - p_1) & \frac{1}{3} * (1 - p_2) & \frac{1}{3} * (1 - p_3 + p_1 + p_2) & 0 \\ \frac{1}{3} * (1 - p_0) & 0 & 0 & 0 & \frac{1}{3} * (2 - p_2 - p_1 + p_0) \\ 0 & \frac{1}{3} * (1 - p_2) & 0 & 0 & \frac{1}{3} * (1 - p_1) \\ 0 & 0 & \frac{1}{3} * (1 - p_1) & 0 & \frac{1}{3} * (1 - p_2) \\ 0 & 0 & 0 & \frac{1}{3} * (1 - p_3) & 0 \end{pmatrix};$$

In[8]:= matrizAleatoria = (γ) * (A) + (1 - γ) * B;

In[9]:= Id = IdentityMatrix[8];
└matriz identidad

In[10]:= autovectores = NullSpace[Id - Transpose[matrizAleatoria]];
└espacio nulo └transposición

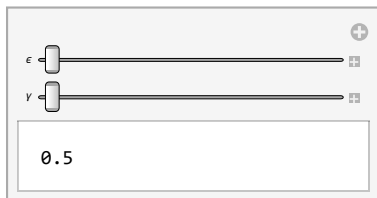
In[11]:= autovectoresNormalizados = autovectores /
 (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
 autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]]);

In[12]:= (γ) * (1 / 2 - ε) + (1 - γ) * $\left(\begin{aligned} &(\text{autovectoresNormalizados}[[1, 1]]) * \left(\frac{1}{3} p_0 + \frac{1}{3} p_0 + \frac{1}{3} p_0 \right) + \\ &(\text{autovectoresNormalizados}[[1, 2]]) * \left(\frac{1}{3} p_2 + \frac{1}{3} p_1 + \frac{1}{3} p_0 \right) + \\ &(\text{autovectoresNormalizados}[[1, 3]]) * \left(\frac{1}{3} p_1 + \frac{1}{3} p_0 + \frac{1}{3} p_2 \right) + \\ &(\text{autovectoresNormalizados}[[1, 4]]) * \left(\frac{1}{3} p_3 + \frac{1}{3} p_1 + \frac{1}{3} p_2 \right) + \\ &(\text{autovectoresNormalizados}[[1, 5]]) * \left(\frac{1}{3} p_0 + \frac{1}{3} p_2 + \frac{1}{3} p_1 \right) + \\ &(\text{autovectoresNormalizados}[[1, 6]]) * \left(\frac{1}{3} p_2 + \frac{1}{3} p_3 + \frac{1}{3} p_1 \right) + \\ &(\text{autovectoresNormalizados}[[1, 7]]) * \left(\frac{1}{3} p_1 + \frac{1}{3} p_2 + \frac{1}{3} p_3 \right) + \\ &(\text{autovectoresNormalizados}[[1, 8]]) * \left(\frac{1}{3} p_3 + \frac{1}{3} p_3 + \frac{1}{3} p_3 \right) \end{aligned} \right);$

```
In[13]:= Simplify[
  (γ) * (1/2 - ε) + (1 - γ) * (
    (autovectoresNormalizados[[1, 1]]) * (1/3 p0 + 1/3 p0 + 1/3 p0) +
    (autovectoresNormalizados[[1, 2]]) * (1/3 p2 + 1/3 p1 + 1/3 p0) +
    (autovectoresNormalizados[[1, 3]]) * (1/3 p1 + 1/3 p0 + 1/3 p2) +
    (autovectoresNormalizados[[1, 4]]) * (1/3 p3 + 1/3 p1 + 1/3 p2) +
    (autovectoresNormalizados[[1, 5]]) * (1/3 p0 + 1/3 p2 + 1/3 p1) +
    (autovectoresNormalizados[[1, 6]]) * (1/3 p2 + 1/3 p3 + 1/3 p1) +
    (autovectoresNormalizados[[1, 7]]) * (1/3 p1 + 1/3 p2 + 1/3 p3) +
    (autovectoresNormalizados[[1, 8]]) * (1/3 p3 + 1/3 p3 + 1/3 p3)
  )
]
Out[13]=
```

$$\frac{(39 + \gamma) (1 + \gamma - 4 \epsilon)}{78 + 8 \gamma^2 - 52 \epsilon + \gamma (74 + 52 \epsilon)}$$

```
In[14]:= Manipulate[
  Manipulate[
    Plot[
      (39 + γ) (1 + γ - 4 ε) / (78 + 8 γ^2 - 52 ε + γ (74 + 52 ε)), {ε, 0, 0.25}, {γ, 0, 1}
    ]
  ], {ε, 0, 0.25}, {γ, 0, 1}
]
Out[14]=
```



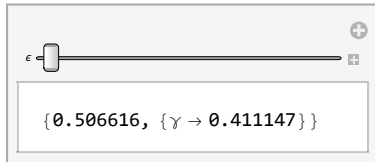
```
In[16]:= Manipulate[
  Plot[
    (39 + γ) (1 + γ - 4 ε) / (78 + 8 γ^2 - 52 ε + γ (74 + 52 ε)), {γ, 0, 1}
  ], {ε, 0, 0.25}
]
Out[16]=
```

¿Donde se da y cuál es el máximo de dicha probabilidad, en función de ϵ ?

```
In[17]:= probabilidad[γ_, ε_] := (39 + γ) (1 + γ - 4 ε) / (78 + 8 γ^2 - 52 ε + γ (74 + 52 ε));
```

```
In[18]:= Manipulate[Module[{maximo}, maximo = FindMaximum[{probabilidad[γ, ε], 0 ≤ γ ≤ 1}, γ];
{maximo[[1]], maximo[[2]]}], {{ε, 0, "ε"}, 0, 0.25, 0.001}]
```

Out[18]=



Representación en el plano de coordenadas ϵ y γ de los puntos donde esta probabilidad es mayor o igual que 0.5

```
In[19]:= RegionPlot[ $\frac{(39 + \gamma) (1 + \gamma - 4 \epsilon)}{78 + 8 \gamma^2 - 52 \epsilon + \gamma (74 + 52 \epsilon)} \geq 0.5$ , {γ, 0, 1},
{ε, 0, 0.007}, PlotPoints → 50, Mesh → None, AxesLabel → {"γ", "ε"},
PlotStyle → Directive[Opacity[0.7], Blue], Frame → True, FrameLabel → {"γ", "ε"}];
```

Estudiemos en qué valor de γ se da el mayor rango de valores posibles de ϵ y cual es ese valor máximo de ϵ

```
In[20]:= ContourPlot[(39 + γ) (1 + γ - 4 ε) / (78 + 8 γ^2 - 52 ε + γ (74 + 52 ε)) - 0.5 == 0,
{γ, 0, 1}, {ε, 0, 0.007}, ContourStyle → Directive[Blue, Thick],
FrameLabel → {"γ", "ε"}, PlotPoints → 50, Axes → True];

In[21]:= FindMaximum[{ε, ((39 + γ) (1 + γ - 4 ε)) / (78 + 8 γ^2 - 52 ε + γ (74 + 52 ε)) == 0.5,
0 ≤ γ ≤ 1, 0 ≤ ε ≤ 0.007}, {{γ, 0.5}, {ε, 0.003}}]

Out[21]= {0.00518633, {γ → 0.474069, ε → 0.00518633}}
```

7.2.9. Estudio del espacio de parámetros

Calculemos el conjunto de puntos (pm, pb) donde la probabilidad de ganar el juego B es menor o igual que 0.5

```

In[ ]:= B =  $\begin{pmatrix} 0 & pm & 1 - pm \\ 1 - pb & 0 & pb \\ pb & 1 - pb & 0 \end{pmatrix}$ ;

In[ ]:= Id = IdentityMatrix[3];
      matriz identidad

In[ ]:= autovectores = NullSpace[Id - Transpose[B]];
      espacio nulo      transposición

In[ ]:= autovectoresNormalizados =
      autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])

Out[ ]:=  $\left\{ \left\{ -\frac{-1 + pb - pb^2}{(1 - pm + pb pm) \left( 1 - \frac{-1 + pb - pb^2}{1 - pm + pb pm} - \frac{-1 + pb - pb pm}{1 - pm + pb pm} \right)}, \right. \right.$ 
 $\left. -\frac{-1 + pb - pb pm}{(1 - pm + pb pm) \left( 1 - \frac{-1 + pb - pb^2}{1 - pm + pb pm} - \frac{-1 + pb - pb pm}{1 - pm + pb pm} \right)}, \frac{1}{1 - \frac{-1 + pb - pb^2}{1 - pm + pb pm} - \frac{-1 + pb - pb pm}{1 - pm + pb pm}} \right\}$ 

In[ ]:= (autovectoresNormalizados[[1, 1]] * (pm) + (1 - (autovectoresNormalizados[[1, 1]])) * (pb)

Out[ ]:=  $-\frac{(-1 + pb - pb^2) pm}{(1 - pm + pb pm) \left( 1 - \frac{-1 + pb - pb^2}{1 - pm + pb pm} - \frac{-1 + pb - pb pm}{1 - pm + pb pm} \right)} + pb \left( 1 + \frac{-1 + pb - pb^2}{(1 - pm + pb pm) \left( 1 - \frac{-1 + pb - pb^2}{1 - pm + pb pm} - \frac{-1 + pb - pb pm}{1 - pm + pb pm} \right)} \right)$ 

In[ ]:= probabilidad1[pm_, pb_] :=
 $-\frac{(-1 + pb - pb^2) pm}{(1 - pm + pb pm) \left( 1 - \frac{-1 + pb - pb^2}{1 - pm + pb pm} - \frac{-1 + pb - pb pm}{1 - pm + pb pm} \right)} + pb \left( 1 + \frac{-1 + pb - pb^2}{(1 - pm + pb pm) \left( 1 - \frac{-1 + pb - pb^2}{1 - pm + pb pm} - \frac{-1 + pb - pb pm}{1 - pm + pb pm} \right)} \right)$ 

In[ ]:= RegionPlot[probabilidad1[pm, pb] ≤ 0.5, {pm, 0, 1},
      representación gráfica de una región
      {pb, 0, 1}, PlotPoints → 50, Mesh → None, AxesLabel → {"pm", "pb"}];
      número de puntos en malla      ning...      etiqueta de ejes

```

Calculemos el conjunto de puntos (p, pm, pb) donde las probabilidades de ganar los juegos A y B son menores o iguales que 0.5

```
In[ ]:= RegionPlot3D[probabilidad1[pm, pb] ≤ 0.5 && p ≤ 0.5, {p, 0, 1}, {pm, 0, 1},
  {pb, 0, 1}, PlotPoints → 50, Mesh → None, AxesLabel → {"p", "pm", "pb"}];
```

[representación de región 3D] [número de puntos en malla] [ninguna etiqueta de ejes]

Power: Infinite expression $\frac{1}{0.}$ encountered. ⓘ

Power: Infinite expression $\frac{1}{0.}$ encountered. ⓘ

Infinity: Indeterminate expression 1 + ComplexInfinity + ComplexInfinity encountered. ⓘ

Power: Infinite expression $\frac{1}{0.}$ encountered. ⓘ

General: Further output of Power::infy will be suppressed during this calculation. ⓘ

Infinity: Indeterminate expression 1 + ComplexInfinity + ComplexInfinity encountered. ⓘ

Infinity: Indeterminate expression 1 + ComplexInfinity + ComplexInfinity encountered. ⓘ

General: Further output of Infinity::indet will be suppressed during this calculation. ⓘ

Calculemos el conjunto de puntos (p, pm, pb) donde la probabilidad de ganar el juego A + B es mayor o igual que 0.5

```
In[ ]:= A =  $\begin{pmatrix} 0 & p & 1-p \\ 1-p & 0 & p \\ p & 1-p & 0 \end{pmatrix}$ ;
```

```
In[ ]:= matrizAleatoria = γ * A + (1 - γ) * B;
```

```
In[ ]:= autovectores = NullSpace[Id - Transpose[matrizAleatoria]];
```

[espacio nulo] [transposición]

In[*]:= **autovectoresNormalizados =**

autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]])

Out[*]=

$$\left\{ \left\{ - \left(\frac{(1 - pb + pb^2 - p \gamma + pb \gamma + 2 p pb \gamma - 2 pb^2 \gamma + p^2 \gamma^2 - 2 p pb \gamma^2 + pb^2 \gamma^2)}{(-1 + pm - pb pm + p \gamma - p pb \gamma - pm \gamma - p pm \gamma + 2 pb pm \gamma - p^2 \gamma^2 + p pb \gamma^2 + p pm \gamma^2 - pb pm \gamma^2)} \left(1 - \frac{1 - pb + pb^2 - p \gamma + pb \gamma + 2 p pb \gamma - 2 pb^2 \gamma + p^2 \gamma^2 - 2 p pb \gamma^2 + pb^2 \gamma^2}{-1 + pm - pb pm + p \gamma - p pb \gamma - pm \gamma - p pm \gamma + 2 pb pm \gamma - p^2 \gamma^2 + p pb \gamma^2 + p pm \gamma^2 - pb pm \gamma^2} - \frac{-(1 - pb)(1 - \gamma) - (1 - p) \gamma - (-pb(1 - \gamma) - p \gamma)(-pm(1 - \gamma) - p \gamma)}{1 - (-(1 - pb)(1 - \gamma)) - (1 - p) \gamma - (-pm(1 - \gamma) - p \gamma)} \right) \right\}, \right. \\ \left. - \left(\frac{-(1 - pb)(1 - \gamma) - (1 - p) \gamma - (-pb(1 - \gamma) - p \gamma)(-pm(1 - \gamma) - p \gamma)}{1 - (-(1 - pb)(1 - \gamma)) - (1 - p) \gamma - (-pm(1 - \gamma) - p \gamma)} \right) \left(1 - \frac{1 - pb + pb^2 - p \gamma + pb \gamma + 2 p pb \gamma - 2 pb^2 \gamma + p^2 \gamma^2 - 2 p pb \gamma^2 + pb^2 \gamma^2}{-1 + pm - pb pm + p \gamma - p pb \gamma - pm \gamma - p pm \gamma + 2 pb pm \gamma - p^2 \gamma^2 + p pb \gamma^2 + p pm \gamma^2 - pb pm \gamma^2} - \frac{-(1 - pb)(1 - \gamma) - (1 - p) \gamma - (-pb(1 - \gamma) - p \gamma)(-pm(1 - \gamma) - p \gamma)}{1 - (-(1 - pb)(1 - \gamma)) - (1 - p) \gamma - (-pm(1 - \gamma) - p \gamma)} \right) \right\}, 1 / \left(1 - \frac{1 - pb + pb^2 - p \gamma + pb \gamma + 2 p pb \gamma - 2 pb^2 \gamma + p^2 \gamma^2 - 2 p pb \gamma^2 + pb^2 \gamma^2}{-1 + pm - pb pm + p \gamma - p pb \gamma - pm \gamma - p pm \gamma + 2 pb pm \gamma - p^2 \gamma^2 + p pb \gamma^2 + p pm \gamma^2 - pb pm \gamma^2} - \frac{-(1 - pb)(1 - \gamma) - (1 - p) \gamma - (-pb(1 - \gamma) - p \gamma)(-pm(1 - \gamma) - p \gamma)}{1 - (-(1 - pb)(1 - \gamma)) - (1 - p) \gamma - (-pm(1 - \gamma) - p \gamma)} \right) \right\} \right\}$$

```

In[ ]:= (γ) * (p) + (1 - γ) *
          (autovectoresNormalizados[[1, 1]] * pm + (1 - (autovectoresNormalizados[[1, 1]])) * (pb))

Out[ ]:=
p γ + (1 - γ) ⎛ - ⎛ (pm (1 - pb + pb² - p γ + pb γ + 2 p pb γ - 2 pb² γ + p² γ² - 2 p pb γ² + pb² γ²)) ⎞ /
⎛ (-1 + pm - pb pm + p γ - p pb γ - pm γ - p pm γ + 2 pb pm γ - p² γ² + p pb γ² + p pm γ² - pb pm γ²)
⎛ 1 - (1 - pb + pb² - p γ + pb γ + 2 p pb γ - 2 pb² γ + p² γ² - 2 p pb γ² + pb² γ²) / (-1 + pm -
pb pm + p γ - p pb γ - pm γ - p pm γ + 2 pb pm γ - p² γ² + p pb γ² + p pm γ² - pb pm γ²) -
- ((1 - pb) (1 - γ)) - (1 - p) γ - (-pb (1 - γ) - p γ) (-pm (1 - γ) - p γ)
1 - ((1 - pb) (1 - γ)) - (1 - p) γ (-pm (1 - γ) - p γ) ⎞ ⎞ ⎞ +
pb ⎛ 1 + (1 - pb + pb² - p γ + pb γ + 2 p pb γ - 2 pb² γ + p² γ² - 2 p pb γ² + pb² γ²) / ⎛ (-1 +
pm - pb pm + p γ - p pb γ - pm γ - p pm γ + 2 pb pm γ - p² γ² + p pb γ² + p pm γ² - pb pm γ²)
⎛ 1 - (1 - pb + pb² - p γ + pb γ + 2 p pb γ - 2 pb² γ + p² γ² - 2 p pb γ² + pb² γ²) / (-1 + pm -
pb pm + p γ - p pb γ - pm γ - p pm γ + 2 pb pm γ - p² γ² + p pb γ² + p pm γ² - pb pm γ²) -
- ((1 - pb) (1 - γ)) - (1 - p) γ - (-pb (1 - γ) - p γ) (-pm (1 - γ) - p γ)
1 - ((1 - pb) (1 - γ)) - (1 - p) γ (-pm (1 - γ) - p γ) ⎞ ⎞ ⎞ ⎞ ⎞

In[ ]:= FullSimplify[(γ) * (p) + (1 - γ) *
[simplifica completamente
          (autovectoresNormalizados[[1, 1]] * pm + (1 - (autovectoresNormalizados[[1, 1]])) * (pb))]

Out[ ]:=
(-pb² (-1 + γ)² (1 + 3 pm (-1 + γ) - 3 p γ) +
3 p γ (1 + p γ (-1 + p γ)) - pm (-1 + γ) (1 + p γ (-2 + 3 p γ)) +
2 pb (-1 + γ) (-1 + p γ (2 - 3 p γ) + pm (-1 + γ) (-1 + 3 p γ))) /
(3 + pb² (-1 + γ)² + 2 pb (-1 + γ) (1 + pm (-1 + γ) - 2 p γ) +
3 p γ (-1 + p γ) - pm (-1 + γ) (-1 + 2 p γ))

In[ ]:= γ = 0.5;

In[ ]:= (-pb² (-1 + γ)² (1 + 3 pm (-1 + γ) - 3 p γ) +
3 p γ (1 + p γ (-1 + p γ)) - pm (-1 + γ) (1 + p γ (-2 + 3 p γ)) +
2 pb (-1 + γ) (-1 + p γ (2 - 3 p γ) + pm (-1 + γ) (-1 + 3 p γ))) /
(3 + pb² (-1 + γ)² + 2 pb (-1 + γ) (1 + pm (-1 + γ) - 2 p γ) +
3 p γ (-1 + p γ) - pm (-1 + γ) (-1 + 2 p γ))

Out[ ]:=
(1.5 p (1 + 0.5 (-1 + 0.5 p) p) - 0.25 pb² (1 - 1.5 p - 1.5 pm) +
0.5 (1 + 0.5 p (-2 + 1.5 p)) pm - 1. pb (-1 + 0.5 (2 - 1.5 p) p - 0.5 (-1 + 1.5 p) pm)) /
(3 + 1.5 (-1 + 0.5 p) p + 0.25 pb² - 1. pb (1 - 1. p - 0.5 pm) + 0.5 (-1 + 1. p) pm)

In[ ]:= probabilidad2[p_, pm_, pb_] :=
(1.5` p (1 + 0.5` (-1 + 0.5` p) p) - 0.25` pb² (1 - 1.5` p - 1.5` pm) + 0.5`
(1 + 0.5` p (-2 + 1.5` p)) pm - 1.` pb (-1 + 0.5` (2 - 1.5` p) p - 0.5` (-1 + 1.5` p) pm)) /
(3 + 1.5` (-1 + 0.5` p) p + 0.25` pb² - 1.` pb (1 - 1.` p - 0.5` pm) + 0.5` (-1 + 1.` p) pm)

```

```

In[ ]:= RegionPlot3D[probabilidad2[p, pm, pb] ≥ 0.5,
  {p, 0, 1}, {pm, 0, 1}, {pb, 0, 1}, PlotPoints → 50, Mesh → None,
  AxesLabel → {"p", "pm", "pb"}, PlotLabel → "Región donde g(p,pm,pb)>0.5"];

```

Calculemos el conjunto de puntos (p, pm, pb) donde la probabilidad de ganar el juego B es menor o igual que 0.5 y la de ganar el juego A + B es mayor o igual que 0.5

```

In[ ]:= RegionPlot3D[probabilidad2[p, pm, pb] ≥ 0.5 && probabilidad1[pm, pb] ≤ 0.5 && p ≤ 0.5,
  {p, 0, 1}, {pm, 0, 1}, {pb, 0, 1}, PlotPoints → 50, Mesh → None,
  AxesLabel → {"p", "pm", "pb"}, PlotLabel → "Región donde g(p,pm,pb)>0.5"];

```

```

Power: Infinite expression 1/0. encountered.
Power: Infinite expression 1/0. encountered.
Infinity: Indeterminate expression 1 + ComplexInfinity + ComplexInfinity encountered.
Power: Infinite expression 1/0. encountered.
General: Further output of Power::infy will be suppressed during this calculation.
Infinity: Indeterminate expression 1 + ComplexInfinity + ComplexInfinity encountered.
Infinity: Indeterminate expression 1 + ComplexInfinity + ComplexInfinity encountered.
General: Further output of Infinity::indet will be suppressed during this calculation.

```

A continuación podemos modificar este volumen en función del parámetro γ (aunque el cálculo es muy costoso computacionalmente)

```

In[ ]:= Clear[γ]

In[ ]:= probabilidad1[pm_, pb_] :=
  -((( -1 + pb - pb^2) pm) / ((1 - pm + pb pm) (1 - (-1 + pb - pb^2) / (1 - pm + pb pm) -
    (-1 + pb - pb pm) / (1 - pm + pb pm)))) + pb (1 + (-1 + pb - pb^2) / ((1 - pm + pb pm)
    (1 - (-1 + pb - pb^2) / (1 - pm + pb pm) - (-1 + pb - pb pm) / (1 - pm + pb pm))))

In[ ]:= probabilidad2[p_, pm_, pb_, γ_] :=
  (-pb^2 (-1 + γ)^2 (1 + 3 pm (-1 + γ) - 3 p γ) + 3 p γ (1 + p γ (-1 + p γ)) - pm (-1 + γ)
    (1 + p γ (-2 + 3 p γ)) + 2 pb (-1 + γ) (-1 + p γ (2 - 3 p γ) + pm (-1 + γ) (-1 + 3 p γ))) /
    (3 + pb^2 (-1 + γ)^2 + 2 pb (-1 + γ) (1 + pm (-1 + γ) - 2 p γ) +
    3 p γ (-1 + p γ) - pm (-1 + γ) (-1 + 2 p γ))

```

```

In[*]:= Manipulate[RegionPlot3D[
  [manipula [representación de región 3D
    probabilidad2[p, pm, pb, γ] ≥ 0.5 && probabilidad1[pm, pb] ≤ 0.5 && p ≤ 0.5, {p, 0, 1},
    {pm, 0, 1}, {pb, 0, 1}, PlotPoints → 50, Mesh → None, AxesLabel → {"p", "pm", "pb"},
    [número de puntos en [malla [ning... [etiqueta de ejes
    PlotLabel → "Región donde  $g(p, pm, pb, \gamma) > 0.5$ ", {γ, 0, 1}];
  [etiqueta de representación

```

7.2.10. Estudio de la mezcla de tres juegos

Comprobemos si la paradoja de Parrondo se puede obtener al mezclar tres juegos

```
In[*]:= p = 1 / 2 - ε;
pm = 1 / 10 - ε;
pb = 3 / 4 - ε;
p1 = 9 / 10 - ε;
p2 = 1 / 4 - ε;
p3 = 1 / 4 - ε;
p4 = 7 / 10 - ε;
```

$$\text{In[*]:= } \mathbf{A} = \begin{pmatrix} 0 & 0 & 1-p & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p & 0 & 0 & 0 & 0 & p & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p & 0 & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & p \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & p \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & p \\ 0 & 0 & 1-p & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p & 0 & 0 & 0 & 0 & p & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p & 0 & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & p \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & p \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & p & 0 \end{pmatrix};$$

$$\text{In[*]:= } \mathbf{B} = \begin{pmatrix} 0 & 0 & 1-pm & 0 & pm & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-pb & 0 & 0 & 0 & 0 & pb & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-pb & 0 & 0 & pb & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-pm & 0 & pm \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-pb & 0 & 0 & 0 & pb \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-pb & 0 & pb & 0 \\ 0 & 0 & 1-pm & 0 & pm & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-pb & 0 & 0 & 0 & 0 & pb & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-pb & 0 & 0 & pb & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-pm & 0 & pm \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-pb & 0 & 0 & 0 & pb \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-pb & 0 & pb & 0 \end{pmatrix};$$

$$\text{In[*]:= } \mathbf{F} = \begin{pmatrix} 0 & 0 & 1-p1 & 0 & p1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p1 & 0 & 0 & 0 & 0 & p1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p1 & 0 & 0 & p1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p2 & 0 & p2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p2 & 0 & 0 & 0 & p2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p2 & 0 & p2 & 0 \\ 0 & 0 & 1-p3 & 0 & p3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-p3 & 0 & 0 & 0 & 0 & p3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-p3 & 0 & 0 & p3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p4 & 0 & p4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p4 & 0 & 0 & 0 & p4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p4 & 0 & p4 & 0 \end{pmatrix};$$

Tenemos tres juegos: el A es el primero presentado en el trabajo, con probabilidad p de ganar. El B es el dependiente del capital en el caso $M = 3$, y el F es el dependiente de la historia

Primero comprobamos que el juego A es justo

```
In[*]:= Id = IdentityMatrix[12];
```

[matriz identidad]

```
In[*]:= autovectores = NullSpace[Id - Transpose[A]];
```

[espacio nulo]

[transposición]

```
In[*]:= autovectoresNormalizados = autovectores /
  (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
   autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]] +
   autovectores[[1, 9]] + autovectores[[1, 10]] + autovectores[[1, 11]] + autovectores[[1, 12]])
```

```
Out[*]=
```

$$\left\{ \left\{ -\frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2 \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2 \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, \frac{(1+2\epsilon)^2}{(-1+2\epsilon)^2 \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, \frac{1+2\epsilon}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, -\frac{1+2\epsilon}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, -\frac{1+2\epsilon}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, -\frac{1+2\epsilon}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, -\frac{1+2\epsilon}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, -\frac{1+2\epsilon}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, -\frac{1+2\epsilon}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, -\frac{1+2\epsilon}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, -\frac{1}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)}, \frac{1}{3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2}}, \frac{1}{3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2}} \right\} \right\}$$

```
In[*]:= Series[autovectoresNormalizados[[1, 1]], {ϵ, 0, 2}]
```

[serie]

```
Out[*]=
```

$$\frac{1}{12} + \frac{\epsilon}{3} + \frac{\epsilon^2}{3} + 0[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 2]], {ϵ, 0, 2}]
```

[serie]

```
Out[*]=
```

$$\frac{1}{12} + \frac{\epsilon}{3} + \frac{\epsilon^2}{3} + 0[\epsilon]^3$$

```

In[ ]:= Series[autovectoresNormalizados[[1, 3]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} + \frac{\epsilon}{3} + \frac{\epsilon^2}{3} + O[\epsilon]^3$$

In[ ]:= Series[autovectoresNormalizados[[1, 4]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} - \frac{\epsilon^2}{3} + O[\epsilon]^3$$

In[ ]:= Series[autovectoresNormalizados[[1, 5]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} - \frac{\epsilon^2}{3} + O[\epsilon]^3$$

In[ ]:= Series[autovectoresNormalizados[[1, 6]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} - \frac{\epsilon^2}{3} + O[\epsilon]^3$$

In[ ]:= Series[autovectoresNormalizados[[1, 7]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} - \frac{\epsilon^2}{3} + O[\epsilon]^3$$

In[ ]:= Series[autovectoresNormalizados[[1, 8]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} - \frac{\epsilon^2}{3} + O[\epsilon]^3$$

In[ ]:= Series[autovectoresNormalizados[[1, 9]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} - \frac{\epsilon^2}{3} + O[\epsilon]^3$$

In[ ]:= Series[autovectoresNormalizados[[1, 10]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} - \frac{\epsilon}{3} + \frac{\epsilon^2}{3} + O[\epsilon]^3$$

In[ ]:= Series[autovectoresNormalizados[[1, 11]], {ϵ, 0, 2}]
Out[ ]:=

$$\frac{1}{12} - \frac{\epsilon}{3} + \frac{\epsilon^2}{3} + O[\epsilon]^3$$


```

```
In[*]:= Series[autovectoresNormalizados[[1, 12]], {ϵ, 0, 2}]
```

```
Out[*]=
```

$$\frac{1}{12} - \frac{\epsilon}{3} + \frac{\epsilon^2}{3} + O[\epsilon]^3$$

La distribución estacionaria para $\epsilon = 0$ es (1/12, 1/12, 1/12, 1/12, 1/12, 1/12, 1/12, 1/12, 1/12, 1/12, 1/12, 1/12)

```
In[*]:= (autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 2]] +
autovectoresNormalizados[[1, 3]] + autovectoresNormalizados[[1, 4]] +
autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]] +
autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 8]] +
autovectoresNormalizados[[1, 9]] + autovectoresNormalizados[[1, 10]] +
autovectoresNormalizados[[1, 11]] + autovectoresNormalizados[[1, 12]]) * (p)
```

```
Out[*]=
```

$$\left(\frac{1}{2} - \epsilon \right) \left(\frac{3}{3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2}} - \frac{6(1+2\epsilon)}{(-1+2\epsilon) \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2 \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2 \left(3 - \frac{6(1+2\epsilon)}{-1+2\epsilon} + \frac{2(1+2\epsilon)^2}{(-1+2\epsilon)^2} - \frac{-1-4\epsilon-4\epsilon^2}{(-1+2\epsilon)^2} \right)} \right)$$

```
In[*]:= Series[(autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 2]] +
autovectoresNormalizados[[1, 3]] + autovectoresNormalizados[[1, 4]] +
autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]] +
autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 8]] +
autovectoresNormalizados[[1, 9]] + autovectoresNormalizados[[1, 10]] +
autovectoresNormalizados[[1, 11]] + autovectoresNormalizados[[1, 12]]) * (p), {ϵ, 0, 2}]
```

```
Out[*]=
```

$$\frac{1}{2} - \epsilon + O[\epsilon]^3$$

Efectivamente el juego A es justo para $\epsilon = 0$.

Trabajemos ahora con B

```
In[*]:= autovectores = NullSpace[Id - Transpose[B]];
```

```
In[*]:= autovectoresNormalizados =
autovectores / (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] +
autovectores[[1, 4]] + autovectores[[1, 5]] + autovectores[[1, 6]] +
autovectores[[1, 7]] + autovectores[[1, 8]] + autovectores[[1, 9]] +
autovectores[[1, 10]] + autovectores[[1, 11]] + autovectores[[1, 12]]);
```

```
In[*]:= (autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 4]] +
          autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 10]]) * pm +
(autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]] +
 autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]] +
 autovectoresNormalizados[[1, 8]] + autovectoresNormalizados[[1, 9]] +
 autovectoresNormalizados[[1, 11]] + autovectoresNormalizados[[1, 12]]) * pb);
```

```
In[*]:= Series[ ( (autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 4]] +
[serie
          autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 10]]) * pm +
(autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]] +
 autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]] +
 autovectoresNormalizados[[1, 8]] + autovectoresNormalizados[[1, 9]] +
 autovectoresNormalizados[[1, 11]] +
          autovectoresNormalizados[[1, 12]]) * pb), {ϵ, 0, 2}]
```

Out[*]=

$$\frac{1}{2} - \frac{147}{169} \epsilon + \frac{1704}{28561} \epsilon^2 + O[\epsilon]^3$$

Efectivamente el juego B es justo para $\epsilon = 0$.

Trabajemos ahora con F

```
In[*]:= autovectores = NullSpace[Id - Transpose[F]];
[espacio nulo [transposición
```

```
In[*]:= autovectoresNormalizados = autovectores /
      (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
      autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]] +
      autovectores[[1, 9]] + autovectores[[1, 10]] + autovectores[[1, 11]] + autovectores[[1, 12]])
```

```
Out[*]=
```

$$\left\{ \left\{ -\frac{-9-42\epsilon-40\epsilon^2}{(-1+4\epsilon)(-9+10\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, \right. \right. \\ -\frac{-9-42\epsilon-40\epsilon^2}{(-1+4\epsilon)(-9+10\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, \\ -\frac{-9-42\epsilon-40\epsilon^2}{(-1+4\epsilon)(-9+10\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, \\ -\frac{2(3+10\epsilon)}{5(-1+4\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, -\frac{2(3+10\epsilon)}{5(-1+4\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, \\ -\frac{2(3+10\epsilon)}{5(-1+4\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, -\frac{2(3+10\epsilon)}{5(-1+4\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, \\ -\frac{2(3+10\epsilon)}{5(-1+4\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, -\frac{2(3+10\epsilon)}{5(-1+4\epsilon)\left(3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}\right)}, \\ \left. \left. 3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}, 3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)}, 3-\frac{12(3+10\epsilon)}{5(-1+4\epsilon)}-\frac{3(-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon)(-9+10\epsilon)} \right\} \right\}$$

```
In[*]:= Series[autovectoresNormalizados[[1, 1]], {ϵ, 0, 2}]
```

```
|serie
```

```
Out[*]=
```

$$\frac{5}{66} + \frac{80\epsilon}{297} + \frac{100\epsilon^2}{2673} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 2]], {ϵ, 0, 2}]
```

```
|serie
```

```
Out[*]=
```

$$\frac{5}{66} + \frac{80\epsilon}{297} + \frac{100\epsilon^2}{2673} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 3]], {ϵ, 0, 2}]
```

```
|serie
```

```
Out[*]=
```

$$\frac{5}{66} + \frac{80\epsilon}{297} + \frac{100\epsilon^2}{2673} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 4]], {ϵ, 0, 2}]
```

```
|serie
```

```
Out[*]=
```

$$\frac{1}{11} + \frac{10\epsilon}{99} - \frac{400\epsilon^2}{891} + O[\epsilon]^3$$

```
In[ ]:= Series[autovectoresNormalizados[[1, 5]], {ϵ, 0, 2}]
```

```
_serie
```

```
Out[ ]:=
```

$$\frac{1}{11} + \frac{10\epsilon}{99} - \frac{400\epsilon^2}{891} + O[\epsilon]^3$$

```
In[ ]:= Series[autovectoresNormalizados[[1, 6]], {ϵ, 0, 2}]
```

```
_serie
```

```
Out[ ]:=
```

$$\frac{1}{11} + \frac{10\epsilon}{99} - \frac{400\epsilon^2}{891} + O[\epsilon]^3$$

```
In[ ]:= Series[autovectoresNormalizados[[1, 7]], {ϵ, 0, 2}]
```

```
_serie
```

```
Out[ ]:=
```

$$\frac{1}{11} + \frac{10\epsilon}{99} - \frac{400\epsilon^2}{891} + O[\epsilon]^3$$

```
In[ ]:= Series[autovectoresNormalizados[[1, 8]], {ϵ, 0, 2}]
```

```
_serie
```

```
Out[ ]:=
```

$$\frac{1}{11} + \frac{10\epsilon}{99} - \frac{400\epsilon^2}{891} + O[\epsilon]^3$$

```
In[ ]:= Series[autovectoresNormalizados[[1, 9]], {ϵ, 0, 2}]
```

```
_serie
```

```
Out[ ]:=
```

$$\frac{1}{11} + \frac{10\epsilon}{99} - \frac{400\epsilon^2}{891} + O[\epsilon]^3$$

```
In[ ]:= Series[autovectoresNormalizados[[1, 10]], {ϵ, 0, 2}]
```

```
_serie
```

```
Out[ ]:=
```

$$\frac{5}{66} - \frac{140\epsilon}{297} + \frac{2300\epsilon^2}{2673} + O[\epsilon]^3$$

```
In[ ]:= Series[autovectoresNormalizados[[1, 11]], {ϵ, 0, 2}]
```

```
_serie
```

```
Out[ ]:=
```

$$\frac{5}{66} - \frac{140\epsilon}{297} + \frac{2300\epsilon^2}{2673} + O[\epsilon]^3$$

```
In[ ]:= Series[autovectoresNormalizados[[1, 12]], {ϵ, 0, 2}]
```

```
_serie
```

```
Out[ ]:=
```

$$\frac{5}{66} - \frac{140\epsilon}{297} + \frac{2300\epsilon^2}{2673} + O[\epsilon]^3$$

```
In[*]:= (autovectoresNormalizados[[1, 1]] +
          autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]]) * (p1) +
          (autovectoresNormalizados[[1, 4]] + autovectoresNormalizados[[1, 5]] +
          autovectoresNormalizados[[1, 6]]) * (p2) +
          (autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 8]] +
          autovectoresNormalizados[[1, 9]]) * (p3) +
          (autovectoresNormalizados[[1, 10]] + autovectoresNormalizados[[1, 11]] +
          autovectoresNormalizados[[1, 12]]) * (p4)
```

```
Out[*]=
```

$$\frac{3 \left(\frac{7}{10} - \epsilon \right)}{3 - \frac{12 (3+10\epsilon)}{5 (-1+4\epsilon)} - \frac{3 (-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon) (-9+10\epsilon)}} - \frac{12 \left(\frac{1}{4} - \epsilon \right) (3+10\epsilon)}{5 (-1+4\epsilon) \left(3 - \frac{12 (3+10\epsilon)}{5 (-1+4\epsilon)} - \frac{3 (-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon) (-9+10\epsilon)} \right)} -$$

$$\frac{3 \left(\frac{9}{10} - \epsilon \right) (-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon) (-9+10\epsilon) \left(3 - \frac{12 (3+10\epsilon)}{5 (-1+4\epsilon)} - \frac{3 (-9-42\epsilon-40\epsilon^2)}{(-1+4\epsilon) (-9+10\epsilon)} \right)}$$

```
In[*]:= Series[ (autovectoresNormalizados[[1, 1]] +
                 serie
                 autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]]) * (p1) +
                 (autovectoresNormalizados[[1, 4]] + autovectoresNormalizados[[1, 5]] +
                 autovectoresNormalizados[[1, 6]]) * (p2) +
                 (autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 8]] +
                 autovectoresNormalizados[[1, 9]]) * (p3) +
                 (autovectoresNormalizados[[1, 10]] + autovectoresNormalizados[[1, 11]] +
                 autovectoresNormalizados[[1, 12]]) * (p4), {epsilon, 0, 2}]
```

```
Out[*]=
```

$$\frac{1}{2} - \frac{10\epsilon}{9} + \frac{100\epsilon^2}{81} + O[\epsilon]^3$$

Efectivamente el juego F es justo para $\epsilon = 0$.

```
In[*]:= matrizAleatoria = (1 / 3) * A + (1 / 3) * B + (1 / 3) * F;
```

```
In[*]:= autovectores = NullSpace[Id - Transpose[matrizAleatoria]];
          espacio nulo      transposición
```

```
In[*]:= autovectoresNormalizados = autovectores /
      (autovectores[[1, 1]] + autovectores[[1, 2]] + autovectores[[1, 3]] + autovectores[[1, 4]] +
      autovectores[[1, 5]] + autovectores[[1, 6]] + autovectores[[1, 7]] + autovectores[[1, 8]] +
      autovectores[[1, 9]] + autovectores[[1, 10]] + autovectores[[1, 11]] + autovectores[[1, 12]])
```

$$\text{Out[*]} = \left\{ \left\{ \frac{770\,349\,830 + 4\,390\,255\,140 \epsilon + 8\,528\,360\,800 \epsilon^2 + 8\,737\,104\,000 \epsilon^3 + 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \right. \right. \\ \frac{1\,083\,073\,410 + 4\,606\,000\,020 \epsilon + 7\,422\,549\,600 \epsilon^2 + 7\,885\,968\,000 \epsilon^3 + 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{815\,639\,100 + 4\,099\,234\,800 \epsilon + 7\,592\,293\,200 \epsilon^2 + 9\,397\,608\,000 \epsilon^3 + 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{1\,436\,517\,020 + 1\,343\,154\,240 \epsilon - 1\,185\,765\,200 \epsilon^2 - 515\,304\,000 \epsilon^3 - 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{555\,358\,430 + 1\,034\,894\,340 \epsilon - 1\,583\,955\,200 \epsilon^2 - 2\,026\,944\,000 \epsilon^3 - 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{1\,239\,681\,641 + 2\,366\,125\,540 \epsilon - 1\,767\,962\,400 \epsilon^2 - 1\,366\,440\,000 \epsilon^3 - 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{600\,647\,700 + 743\,874\,000 \epsilon - 2\,520\,022\,800 \epsilon^2 - 1\,366\,440\,000 \epsilon^3 - 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{926\,958\,061 + 2\,150\,380\,660 \epsilon - 662\,151\,200 \epsilon^2 - 515\,304\,000 \epsilon^3 - 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{1\,703\,951\,330 + 1\,849\,919\,460 \epsilon - 1\,355\,508\,800 \epsilon^2 - 2\,026\,944\,000 \epsilon^3 - 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{1\,190\,201\,410 - 3\,069\,516\,780 \epsilon + 2\,154\,092\,000 \epsilon^2 - 4\,634\,064\,000 \epsilon^3 + 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \frac{922\,767\,100 - 3\,576\,282\,000 \epsilon + 2\,323\,835\,600 \epsilon^2 - 3\,122\,424\,000 \epsilon^3 + 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3}, \\ \left. \left. \frac{877\,477\,830 - 3\,285\,261\,660 \epsilon + 3\,259\,903\,200 \epsilon^2 - 3\,782\,928\,000 \epsilon^3 + 6\,058\,080\,000 \epsilon^4}{12\,122\,622\,862 + 12\,652\,777\,760 \epsilon + 22\,205\,668\,800 \epsilon^2 + 6\,663\,888\,000 \epsilon^3} \right\} \right\}$$

```
In[*]:= Series[autovectoresNormalizados[[1, 1]], {ϵ, 0, 2}]
      serie
```

$$\text{Out[*]} = \frac{385\,174\,915}{6\,061\,311\,431} + \frac{1\,552\,655\,076\,204\,758\,210 \epsilon}{5\,248\,499\,466\,224\,466\,823} + \frac{1\,264\,970\,225\,697\,224\,821\,731\,865\,200 \epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 2]], {ϵ, 0, 2}]
      serie
```

$$\text{Out[*]} = \frac{541\,536\,705}{6\,061\,311\,431} + \frac{1\,504\,746\,928\,226\,039\,130 \epsilon}{5\,248\,499\,466\,224\,466\,823} + \frac{678\,956\,576\,037\,285\,230\,600\,910\,000 \epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

```
In[*]:= Series[autovectoresNormalizados[[1, 3]], {ϵ, 0, 2}]
      serie
```

$$\text{Out[*]} = \frac{407\,819\,550}{6\,061\,311\,431} + \frac{1\,406\,192\,044\,232\,842\,200 \epsilon}{5\,248\,499\,466\,224\,466\,823} + \frac{1\,015\,313\,408\,127\,725\,979\,123\,479\,400 \epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 4]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{718\,258\,510}{6\,061\,311\,431} - \frac{67\,620\,653\,767\,901\,440\,\epsilon}{5\,248\,499\,466\,224\,466\,823} - \frac{1\,369\,894\,256\,201\,715\,410\,265\,277\,800\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 5]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{277\,679\,215}{6\,061\,311\,431} + \frac{197\,100\,249\,782\,353\,010\,\epsilon}{5\,248\,499\,466\,224\,466\,823} - \frac{1\,153\,317\,768\,874\,631\,516\,780\,308\,800\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 6]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{1\,239\,681\,641}{12\,122\,622\,862} + \frac{464\,222\,545\,315\,749\,690\,\epsilon}{5\,248\,499\,466\,224\,466\,823} - \frac{1\,933\,650\,510\,091\,479\,045\,539\,875\,200\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 7]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{300\,323\,850}{6\,061\,311\,431} + \frac{50\,637\,217\,810\,437\,000\,\epsilon}{5\,248\,499\,466\,224\,466\,823} - \frac{1\,402\,974\,586\,444\,130\,359\,388\,694\,600\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 8]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{926\,958\,061}{12\,122\,622\,862} + \frac{512\,130\,693\,294\,468\,770\,\epsilon}{5\,248\,499\,466\,224\,466\,823} - \frac{1\,347\,636\,860\,431\,539\,454\,408\,920\,000\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 9]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{851\,975\,665}{6\,061\,311\,431} + \frac{30\,934\,230\,225\,295\,490\,\epsilon}{5\,248\,499\,466\,224\,466\,823} - \frac{1\,706\,251\,088\,292\,156\,158\,787\,847\,200\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 10]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{595\,100\,705}{6\,061\,311\,431} - \frac{1\,866\,783\,865\,103\,188\,070\,\epsilon}{5\,248\,499\,466\,224\,466\,823} + \frac{1\,677\,371\,459\,574\,345\,191\,353\,714\,800\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 11]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{461\,383\,550}{6\,061\,311\,431} - \frac{1\,965\,338\,749\,096\,385\,000\,\epsilon}{5\,248\,499\,466\,224\,466\,823} + \frac{2\,013\,728\,291\,664\,785\,939\,876\,284\,200\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

In[$\ast$]:= Series[autovectoresNormalizados[[1, 12]], { ϵ , 0, 2}]

serie

$$\text{Out}[\ast]= \frac{438\,738\,915}{6\,061\,311\,431} - \frac{1\,818\,875\,717\,124\,468\,990\,\epsilon}{5\,248\,499\,466\,224\,466\,823} + \frac{2\,263\,385\,109\,234\,284\,782\,484\,670\,000\,\epsilon^2}{4\,544\,684\,258\,603\,394\,166\,590\,021\,959} + O[\epsilon]^3$$

```

In[ ]:= (1 / 3) * (1 / 2 - ε) +
(1 / 3) * ((autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 4]] +
autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 10]]) * pm +
(autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]] +
autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]] +
autovectoresNormalizados[[1, 8]] + autovectoresNormalizados[[1, 9]] +
autovectoresNormalizados[[1, 11]] + autovectoresNormalizados[[1, 12]]) * pb) +
(1 / 3) * ((autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 2]] +
autovectoresNormalizados[[1, 3]]) * p1 + (autovectoresNormalizados[[1, 4]] +
autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]]) * p2 +
(autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 8]] +
autovectoresNormalizados[[1, 9]]) * p3 + (autovectoresNormalizados[[1, 10]] +
autovectoresNormalizados[[1, 11]] + autovectoresNormalizados[[1, 12]]) * p4);

In[ ]:= N[Series[ (1 / 3) * (1 / 2 - ε) +

$$\begin{aligned} & (1 / 3) * ((\text{autovectoresNormalizados}[[1, 1]] + \text{autovectoresNormalizados}[[1, 4]] + \\ & \quad \text{autovectoresNormalizados}[[1, 7]] + \text{autovectoresNormalizados}[[1, 10]]) * pm + \\ & \quad (\text{autovectoresNormalizados}[[1, 2]] + \text{autovectoresNormalizados}[[1, 3]] + \\ & \quad \text{autovectoresNormalizados}[[1, 5]] + \text{autovectoresNormalizados}[[1, 6]] + \\ & \quad \text{autovectoresNormalizados}[[1, 8]] + \text{autovectoresNormalizados}[[1, 9]] + \\ & \quad \text{autovectoresNormalizados}[[1, 11]] + \text{autovectoresNormalizados}[[1, 12]]) * pb) + \\ & (1 / 3) * ((\text{autovectoresNormalizados}[[1, 1]] + \text{autovectoresNormalizados}[[1, 2]] + \\ & \quad \text{autovectoresNormalizados}[[1, 3]]) * p1 + (\text{autovectoresNormalizados}[[1, 4]] + \\ & \quad \text{autovectoresNormalizados}[[1, 5]] + \text{autovectoresNormalizados}[[1, 6]]) * p2 + \\ & \quad (\text{autovectoresNormalizados}[[1, 7]] + \text{autovectoresNormalizados}[[1, 8]] + \\ & \quad \text{autovectoresNormalizados}[[1, 9]]) * p3 + \\ & \quad (\text{autovectoresNormalizados}[[1, 10]] + \text{autovectoresNormalizados}[[1, 11]] + \\ & \quad \text{autovectoresNormalizados}[[1, 12]]) * p4), \{\epsilon, 0, 2\}] ]
\end{aligned}$$

Out[ ]:=
0.513256 - 0.96357 (ε + 0.) + 0.329533 (ε + 0.)2 + 0 (ε + 0.)3

```

Observamos que efectivamente se da la paradoja

Estudiamos hasta qué valor de ϵ se da

```
In[ ]:= Solve[(1/3)*(1/2 - ε) +
```

```
resuelve
```

```
(1/3)*((autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 4]] +
autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 10]]) * pm +
(autovectoresNormalizados[[1, 2]] + autovectoresNormalizados[[1, 3]] +
autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]] +
autovectoresNormalizados[[1, 8]] + autovectoresNormalizados[[1, 9]] +
autovectoresNormalizados[[1, 11]] + autovectoresNormalizados[[1, 12]]) * pb) +
(1/3)*((autovectoresNormalizados[[1, 1]] + autovectoresNormalizados[[1, 2]] +
autovectoresNormalizados[[1, 3]]) * p1 + (autovectoresNormalizados[[1, 4]] +
autovectoresNormalizados[[1, 5]] + autovectoresNormalizados[[1, 6]]) * p2 +
(autovectoresNormalizados[[1, 7]] + autovectoresNormalizados[[1, 8]] +
autovectoresNormalizados[[1, 9]]) * p3 +
(autovectoresNormalizados[[1, 10]] + autovectoresNormalizados[[1, 11]] +
autovectoresNormalizados[[1, 12]]) * p4) == 0.5, ε]
```

*** Solve: Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

```
Out[ ]:=
```

```
{ {ε → -0.217312 - 0.756202 i}, {ε → -0.217312 + 0.756202 i}, {ε → 0.0138217} }
```


Bibliografía

- [1] Abbott, D. «Asymmetry and Disorder: A Decade of Parrondo's Paradox». En: *Fluctuation and Noise Letters* 9 (2010), págs. 129-156. DOI: [10.1142/S0219477510000010](https://doi.org/10.1142/S0219477510000010).
- [2] Abbott, D. y Harmer, G.P. «Parrondo's paradox». En: *Statistical Science* 14.2 (1999), págs. 206-213. DOI: [10.1214/ss/1009212247](https://doi.org/10.1214/ss/1009212247).
- [3] *Adelaide Parrondo's Paradox Group*. Recuperado el 30-06-2024. 2006. URL: <http://www.eleceng.adelaide.edu.au/groups/parrondo/index.html>.
- [4] Allison, A. y Abbott, D. «Control systems with stochastic feedback». En: *Chaos* 11 (2001), págs. 715-724. DOI: [10.1063/1.1397769](https://doi.org/10.1063/1.1397769).
- [5] Breuer, S. y Mielke, A. «Multi player Parrondo games with rigid coupling». En: *Physica A: Statistical Mechanics and its Applications* 622(C) (2023). DOI: [10.1016/j.physa.2023.128890](https://doi.org/10.1016/j.physa.2023.128890).
- [6] Brown, R. «A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies». En: *Philosophical Magazine* 4 (1828), págs. 161-173. DOI: [10.1080/14786442808674769](https://doi.org/10.1080/14786442808674769).
- [7] Cao, F.J., Dinis, L. y Parrondo, J.M.R. «Feedback Control in a Collective Flashing Ratchet». En: *Physical Review Letters* 93.4 (2004). DOI: [10.1103/PhysRevLett.93.040603](https://doi.org/10.1103/PhysRevLett.93.040603).
- [8] Cheong, K. H., Wen, T. y Lai, J. «Relieving Cost of Epidemic by Parrondo's Paradox: A COVID-19 Case Study». En: *Advanced Science* 7 (2020). DOI: [10.1002/advs.202002324](https://doi.org/10.1002/advs.202002324).
- [9] Cheong, K.H., Wen, T., Benier, S., Koh, J.M. y Koomin, E.V. «Alternating lysis and lysogeny is a winning strategy in bacteriophages due to Parrondo's paradox». En: *Proceedings of the National Academy of Sciences (PNAS)* 119.13 (2022). DOI: [10.1073/pnas.2115145119](https://doi.org/10.1073/pnas.2115145119).
- [10] Cleuren, C. y van den Broeck, B. «Parrondo Games with Strategy». En: *Proceedings of SPIE* 5471 (2004), págs. 109-118. DOI: [10.1117/12.544274](https://doi.org/10.1117/12.544274).
- [11] Di Crescenzo, A. «A Parrondo Paradox in reliability theory». En: *Mathematical Scientist* 32.1 (2007), págs. 17-22. DOI: [10.48550/arXiv.math/0602308](https://doi.org/10.48550/arXiv.math/0602308).
- [12] Dinis, L. «Optimización y control de juegos de azar y motores brownianos colectivos». Tesis doct. Universidad Complutense de Madrid, 2006. URL: <https://docta.ucm.es/entities/publication/14af8fab-8499-48af-8907-b39a157a5642>.

- [13] Ejlali, N., Pezeshk, H., Chaubey, Y.P. y Sadeghi, M. «Parrondo's paradox for games with three players and its potential application in combination therapy for type II diabetes». En: *Physica A: Statistical Mechanics and its Applications* 556.13 (2020). DOI: [10.1016/j.physa.2020.124707](https://doi.org/10.1016/j.physa.2020.124707).
- [14] Ethier, S.N. y Lee, J. «Parrondo games with spatial dependence». En: *Fluctuation and Noise Letters* 11.2 (2012). DOI: [10.1142/S0219477512500046](https://doi.org/10.1142/S0219477512500046).
- [15] Ethier, S.N. y Lee, J. «The flashing Brownian ratchet and Parrondo's paradox». En: *Royal Society Open Science* (2018). DOI: [10.1098/rsos.171685](https://doi.org/10.1098/rsos.171685).
- [16] Fernández, E.S. «La Paradoja de Parrondo y su relación con el efecto ratchet». Trabajo Fin de Grado. Universidad de Sevilla, 2014. URL: <https://idus.us.es/handle/11441/50132>.
- [17] Guo, J., Xie, N. y Ye, Y. «The theoretical analysis and computer simulation on Parrondo's history dependent games». En: *Procedia Engineering* 15 (2011), págs. 4597-4602. DOI: [10.1016/j.proeng.2011.08.863](https://doi.org/10.1016/j.proeng.2011.08.863).
- [18] Harmer, G.P. y Abbott, D. «A review of Parrondo's Paradox». En: *Fluctuation and Noise Letters* 2.2 (2002), R71-R107. DOI: [10.1142/S0219477502000701](https://doi.org/10.1142/S0219477502000701).
- [19] Harmer, G.P., Abbott, D. y Parrondo, J.M.R. «Parrondo's capital and history-dependent games». En: *Advances in Dynamic Games*. Ed. por A. S. Nowak y K. Szajowski. Vol. 7. Annals of the International Society of Dynamic Games. Birkhäuser Boston, 2005. DOI: [10.1007/0-8176-4429-6_33](https://doi.org/10.1007/0-8176-4429-6_33).
- [20] Harmer, G.P., Abbott, D., Taylor, P.G. y Parrondo, J.M.R. «Brownian ratchets and Parrondo's games». En: *Chaos* 11 (2001), págs. 705-714. DOI: [10.1063/1.1395623](https://doi.org/10.1063/1.1395623).
- [21] Jia, S., Lai, J.W., Koh, J.M., Xie, N.G. y Cheong, K.H. «Parrondo effect: Exploring the nature-inspired framework on periodic functions». En: *Physica A: Statistical Mechanics and its Applications* 556 (2020). DOI: [10.1016/j.physa.2020.124714](https://doi.org/10.1016/j.physa.2020.124714).
- [22] Jia, S., Weijia, J., Koh, J.M. y Cheong, K.H. «Periodic noise-induced framework for history-dependent Parrondo's switching». En: *Nonlinear Dyn* 108 (2022), págs. 1797-1804. DOI: [10.1007/s11071-022-07277-x](https://doi.org/10.1007/s11071-022-07277-x).
- [23] Kay, R.J. y Johnson, N.F. «Winning combinations of history-dependent games». En: *Physics Review E* 67.5 (2003). DOI: [10.1103/PhysRevE.67.056128](https://doi.org/10.1103/PhysRevE.67.056128).
- [24] Kocarev L. y Tasev Z. «Lyapunov exponents, noise-induced synchronization, and Parrondo's paradox». En: *Physical Review E* 65 (2002). DOI: [10.1103/PhysRevE.65.046215](https://doi.org/10.1103/PhysRevE.65.046215).
- [25] Koh, J.M. y Cheong, K.H. «New doubly-anomalous Parrondo's games suggest emergent sustainability and inequality». En: *Nonlinear Dyn* 96 (2019), págs. 257-266. DOI: [10.1007/s11071-019-04788-y](https://doi.org/10.1007/s11071-019-04788-y).
- [26] Korilis, Y.A., Lazar, A.A. y Orda, A. «Avoiding the Braess paradox in non-cooperative networks». En: *Journal of Applied Probability* 36.1 (1999), págs. 211-222. DOI: [10.1239/jap/1032374242](https://doi.org/10.1239/jap/1032374242).
- [27] Lai, J.2. y Cheong, H.K. «Social dynamics and Parrondo's paradox: a narrative review». En: *Nonlinear Dyn* 101 (2020), págs. 1-20. DOI: [10.1007/s11071-020-05738-9](https://doi.org/10.1007/s11071-020-05738-9).

- [28] Lai, J.W. y Cheong, K.H. «Parrondo's paradox from classical to quantum: A review». En: *Nonlinear Dyn* 100 (2020), págs. 849-861. DOI: [10.1007/s11071-020-05496-8](https://doi.org/10.1007/s11071-020-05496-8).
- [29] Lai, J.W. y Cheong, K.H. «Evaluation of single-prioritization voting systems in controlled collective Parrondo's games». En: *Nonlinear Dyn* 107 (2022), págs. 2965-2974. DOI: [10.1007/s11071-021-07087-7](https://doi.org/10.1007/s11071-021-07087-7).
- [30] Mihailovic, Z. y Rajkovic, M. «One Dimensional Asynchronous Cooperative Parrondo's Games». En: *Fluctuation and Noise Letters* 3.4 (2003), págs. L384-L398. DOI: [10.1142/S0219477503001464](https://doi.org/10.1142/S0219477503001464).
- [31] Mihailovic, Z. y Rajkovic, M. «Cooperative Parrondo's games on a two-dimensional lattice». En: *Physica A: Statistical Mechanics and its Applications* 365 (2005). DOI: [10.1016/j.physa.2006.01.032](https://doi.org/10.1016/j.physa.2006.01.032).
- [32] *Parrondo Paradox Blog*. Recuperado el 30-06-2024. 2011. URL: <http://parrondoparadox.blogspot.com>.
- [33] Parrondo, J.M.R. «Juegos de azar paradójicos.» En: *Gaceta de la Real Sociedad Matemática Española* 4 (2001), págs. 67-77. URL: <http://dml.mathdoc.fr/item/urn:eudml:doc:43239>.
- [34] Parrondo, J.M.R. y Dinis, L. «Brownian motion and gambling: from ratchets to paradoxical games». En: *Contemporary Physics* 45 (2004), págs. 147-157. DOI: [10.1080/00107510310001644836](https://doi.org/10.1080/00107510310001644836).
- [35] Parrondo, J.M.R. y Dinis, L. «Inefficiency of voting in Parrondo games». En: *Physica A: Statistical Mechanics and its Applications* 343 (2004), págs. 701-711. DOI: [10.1016/j.physa.2004.06.076](https://doi.org/10.1016/j.physa.2004.06.076).
- [36] Parrondo, J.M.R., Dinis, L. Buceta, J. y Lindenberg, K. «Paradoxical games, ratchets, and related phenomena». En: *Advances in Condensed Matter and Statistical Physics*. Ed. por E. Korutcheva y R. Cuerno. New York: Nova Science Publishers, 2004.
- [37] Parrondo, J.M.R., Dinis, L., García-Toraño, E. y Sotillo, B. «Collective decision making and paradoxical games». En: *European Journal of Physics. Special Topics* 143 (2007), págs. 39-46. DOI: [10.48550/arXiv.1410.0241](https://doi.org/10.48550/arXiv.1410.0241).
- [38] Parrondo, J.M.R., Harmer, G.P. y Derek Abbott, D. «New Paradoxical Games Based on Brownian Ratchets». En: *Physical Review Letters* 85.24 (2000), págs. 5226-5229. DOI: [10.1103/PhysRevLett.85.5226](https://doi.org/10.1103/PhysRevLett.85.5226).
- [39] Pearce, C.E.M. «On Parrondo's Paradoxical Games». En: *Proceedings of Unsolved problems of Noise and Fluctuations - UPoN'99: Second International Conference*. Ed. por Derek Abbott y Laszlo B. Kish. 2000, págs. 201-206. DOI: [10.1063/1.59975](https://doi.org/10.1063/1.59975).
- [40] Python Software Foundation. *Python, Version 3.10.12*. Utilizado en Google Colab. 2023. URL: <https://colab.research.google.com/>.
- [41] Redondo, C.D. «Paradoja de Parrondo y aplicaciones». Trabajo Fin de Grado. Universidad de Valladolid, 2022. URL: <https://uvadoc.uva.es/handle/10324/57943>.

- [42] Reed, F.A. «Two-locus epistasis with sexually antagonistic selection: a genetic Parrondo's paradox». En: *Genetics* 176.3 (2007), págs. 1923-1929. DOI: [10.1534/genetics.106.069997](https://doi.org/10.1534/genetics.106.069997).
- [43] Rivas, M.J. *Cadenas de Markov*. Asignatura Modelos probabilísticos y estadísticos, Máster Universitario en Modelización Matemática, curso 2023/-2024. Universidad de Salamanca.
- [44] Spurgin, R. y Tamarkin, M. «Switching investments can be a bad idea when Parrondo's paradox applies». En: *The Journal of Behavioral Finance* 6.1 (2005), págs. 15-18. DOI: [10.1207/s15427579jpfm0601_3](https://doi.org/10.1207/s15427579jpfm0601_3).
- [45] Tang, T.M., Allison A. y Abbott, D. «Investigation of chaotic switching strategies in Parrondo's games». En: *Fluctuation and Noise Letters* 4.4 (2004), págs. L585-L596. DOI: [10.1142/S021947750400221X](https://doi.org/10.1142/S021947750400221X).
- [46] Taylor, H.M. y Karlin, S. *An introduction to stochastic modeling*. 3.^a ed. Academic Press, Inc., 1984.
- [47] Toral, R. «Cooperative Parrondo's games». En: *Fluctuation and Noise Letters* 1 (2001), págs. L7-L12. DOI: [10.1142/S021947750100007X](https://doi.org/10.1142/S021947750100007X).
- [48] Wolfram Research, Inc. *Mathematica, Version 13.3*. 2023. URL: <https://www.wolfram.com/mathematica/>.
- [49] Xie, N.G., Chen, Y., Ye, Y., Xu, G., Wang, L.G. y Wang, C. «Theoretical analysis and numerical simulation of Parrondo's paradox game in space». En: *Chaos Solitons and Fractals* 44.6 (2011), págs. 401-414. DOI: [10.1016/j.chaos.2011.01.014](https://doi.org/10.1016/j.chaos.2011.01.014).
- [50] Ye, Y., Hang, X.R., Koh, J.M., Miszczak, J.A., Cheong, K.H. y Xie, N.G. «Ratcheting based on neighboring niches determines lifestyle». En: *Nonlinear Dyn* 98 (2019), págs. 1821-1830. DOI: [10.1007/s11071-019-05290-1](https://doi.org/10.1007/s11071-019-05290-1).
- [51] Ye, Y., Zhang, X.S., Liu, L. y Xie, N.G. «Effects of group interactions on the network Parrondo's games». En: *Physica A: Statistical Mechanics and its Applications* 583 (2021). DOI: [10.1016/j.physa.2021.126271](https://doi.org/10.1016/j.physa.2021.126271).