Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Matemáticas

INTELIGENCIA ARTIFICIAL APLICADA A LA TEORÍA DE JUEGOS

Autor: Alejandro Gutiérrez Mielgo
Tutores: Víctor Gatón Bustillo, Cesáreo Jesús González Fernández
2023

Más vale mal acuerdo que buen pleito.
Refrán castellano

Quiero agradecer a mi familia todo el apoyo recibido desde que comencé con este trabajo así como sus ánimos, que me han alentado a lo largo del proceso. De igual manera, quiero dar las gracias a mis amigos por acompañarme y apoyarme, en particular a Paula, que ha estado a mi lado desde el inicio. Por último, quiero agradecer a mis tutores su trabajo y su ayuda; tanto a Víctor y a Cesáreo como a Óscar, que ha ejercido como tal aunque su nombre no aparezca recogido junto a los demás.

Índice general

Introducción **1**

1. Teoría de Juegos **4**
1.1. Conceptos fundamentales 4
1.2. Soluciones de un juego 9
1.2.1. Soluciones mediante argumentos de dominación 9
1.2.2. Soluciones mediante argumentos de equilibrio 12
1.2.3. Equilibrios en estrategias mixtas 16
1.3. Dilema del prisionero 21
1.4. Dilema del prisionero finitamente repetido 22
1.5. Dilema del prisionero infinitamente repetido 28
1.5.1. Perfiles de estrategias que son EN 31

2. Autómatas **36**
2.1. Fundamentos de las máquinas secuenciales estocásticas 36
2.1.1. Introducción. Definiciones y autómatas deterministas 36
2.1.2. Autómatas secuenciales estocásticos 38
2.2. Teoría de estados 43
2.2.1. Nociones sobre minimización de estados 45
2.2.2. Recubrimientos 51
2.3. Minimización de estados en ISSM mediante recubrimientos 52

3. Teoría de juegos evolutiva **59**
3.1. Introducción 59
3.2. Conceptos y resultados fundamentales. 60
3.3. Equilibrios refinados en teoría de juegos evolutiva. 63

4. Implementación **66**
4.1. Descripción general. 66
4.2. Algoritmo genético. 67
4.3. Función `crossover`. 69
4.4. Otras disquisiciones interesantes. 77

A. Código Python **79**

Bibliografía **92**

Abstract

The prisoner's dilemma and its repeated version are paradigmatic examples within game theory. This work provides an introduction to game theory, focusing on the prisoner's dilemma and its finite and infinitely iterated variants. Then, it offers a mathematical description of the concept of stochastic automata and its properties. These automata represent a dense subset of possible strategies in a game, providing a variety of heuristic approaches in game theory. After describing the methodology that will be applied and presenting the results that justify its validity, this work concludes by providing an implementation for the infinitely repeated prisoner's dilemma. No publications have been found in which this methodology is applied to the mentioned game using stochastic automata, thus, the proposed implementation makes use of original ideas to suggest a way to do so.

Resumen

El dilema del prisionero es un ejemplo paradigmático dentro de la teoría de juegos al igual que su versión repetida. En este trabajo se presenta una introducción a la teoría de juegos, centrándose en el dilema del prisionero y sus variantes finita e infinitamente iteradas. Se proporciona una descripción matemática del concepto de autómata en su versión estocástica y se profundiza en sus propiedades. Dichos autómatas permiten representar un subconjunto denso de las estrategias posibles de un juego, lo cual permite desarrollar una metodología heurística para el análisis de juegos. Tras exponer los resultados que justifican matemáticamente la validez de esta metodología, se concluye dando una implementación para el dilema del prisionero repetido infinitamente. No se han encontrado publicaciones en las que se aplique esta metodología a dicho juego utilizando autómatas estocásticos, así pues, la implementación dada utiliza ideas propias para sugerir un modo de hacerlo.

Introducción

Este trabajo tiene como objetivo dar las descripciones matemáticas que justifican el uso de autómatas finitos como herramienta en la búsqueda de soluciones dentro de la teoría de juegos. Un autómata representa la idea de una máquina programada para realizar automáticamente una tarea en particular. En este caso los autómatas serán virtuales y representarán estrategias del juego elegido. Por otro lado, la teoría de juegos es el área de las matemáticas que modela y estudia los conflictos entre varios agentes o jugadores que buscan obtener el mayor beneficio posible. En particular, en este trabajo se toma como modelo el arquetípico dilema del prisionero en su versión repetida infinitamente.

La metodología propuesta consiste en considerar autómatas finitos que representen las posibles estrategias empleables por un jugador para un juego dado. Estos autómatas se definen de manera aleatoria. Para ello se conforma una población de autómatas creados aleatoriamente que evoluciona iterativamente mediante un algoritmo genético. Cuando eventualmente se alcance una población homogénea, los individuos determinan un perfil de estrategias que constituye una posible solución del juego.

Un autómata finito está constituido, esencialmente, por un conjunto de estados con una acción asociada a cada uno y un procedimiento de transiciones entre ellos, determinados por un input, que en este caso será la acción de los rivales en el juego. La metodología anteriormente descrita se explora en un amplio abanico de publicaciones, pero en ningún documento de toda la bibliografía consultada se considera el uso de autómatas estocásticos en juegos repetidos semejantes al propuesto en este trabajo.

El primer capítulo del trabajo se centra en describir el marco de la teoría de juegos y exponer los resultados más importantes relativos al tipo de juego que se está manejando. Es por ello que se usa constantemente como hilo conductor el dilema del prisionero.

En el segundo capítulo se da una descripción formal matemática de los autómatas estocásticos. Este tratamiento formal permite obtener resultados muy potentes, como la equivalencia entre distintas descripciones de un mismo autómata o la reducción de un autómata dado a otro con menos estados que represente el mismo algoritmo. La busqueda documental llevada a cabo en este trabajo ha sido infructuosa en el sentido de dar respuesta a un importante problema que, por lo menos en este documento, se considera abierto: el problema de determinar si es decidible que, dado un autómata estocástico como los que utilizaremos en la práctica, existe otro autómata que represente el mismo algoritmo y tenga un número menor de estados. Cabe reseñar que sí que se dan resultados que permiten garantizar la decidibilidad del problema y la obtención de dicho autómata bajo ciertas condiciones.

En el tercer capítulo, se exponen los principales resultados y definiciones de la teoría de juegos evolutiva. Esta rama de la teoría de juegos permite relacionar la teoría de juegos con los autómatas siguiendo la metodología descrita anteriormente. Se considera necesario reseñar que este tercer capítulo no contiene demostraciones, mostrando la teoría sin entrar en detalles dado que se prevé continuar explorando este campo con más profundidad en un próximo Trabajo Fin de Máster.

Finalmente, el trabajo concluye con un capítulo dedicado a la implementación de la metodología propuesta dando una breve descripción de la misma. En el futuro Trabajo Fin de Máster se tratará de extender esta metodología a un conjunto de juegos lo más general posible, con lo cual no se detalla aquí la descripción con excesiva profundidad, permitiendo una mayor legibilidad del trabajo sin dilatarlo excesivamente.

Más allá de las vías de investigación futura ya descritas, desde un punto de vista más ambicioso y largoplacista, los objetivos buscados responden a la necesidad de conseguir computacionalmente soluciones interpretables de problemas complejos. Actualmente, el desarrollo del aprendizaje automático se adhiere fundamentalmente a las redes neuronales que trabajan a modo de caja negra, mientras que los autómatas permiten identificar y comprender la estrategia llevada a cabo por un jugador.

Por otro lado, la solución al problema sobre decidibilidad abriría la puerta a la obtención del conjunto de las soluciones de un juego, al menos de manera heurística, siendo estas interpretables y analizables. Esto conduciría a una posible interpretación de la estructura del conjunto de soluciones así como a otros análisis sobre el juego estudiado.

Una de las aplicaciones podría ser la modificación de un juego en base a sus soluciones, buscando que el desarrollo normal del mismo conduzca a unos resultados particulares. O, dándole la vuela, aplicado a la economía, puede permitir observar cómo afectan a los diferentes agentes los cambios en las regulaciones. La viabilidad de estos análisis no requiere de la metodología descrita en este trabajo pues, por ejemplo en [24], se utilizan redes neuronales para reducir el número de empates en las partidas de ajedrez. Aun así, la información de la que se dispone sobre jugadas óptimas se enriquece notablemente cuando se puedan determinar las soluciones mediante autómatas.

Para finalizar con esta introducción, se destaca que los dos primeros capítulos son fruto de una intensa documentación y recopilación de resultados entre los que se presentan algunas demostraciones originales. En el Capítulo 4, tomando como hilo conductor la publicación [12], se generalizan sus resultados aplicando lo expuesto en capítulos anteriores junto a ideas originales propias.

1. Teoría de Juegos

1.1. Conceptos fundamentales

Antes de comenzar, se indica que la mayor parte de enunciados y resultados descritos en este capítulo se extraen de [3] y [5]. Ahora bien, apartando el concepto de juego de una definición coloquial en pos de encontrar una representación formal y matemática del concepto, podemos entender que un juego es una situación de conflicto y/o cooperación en la que interactúan individuos racionales analizando los comportamientos y resultados que son de esperar. En este marco, los individuos, denominados jugadores, buscan tomar las decisiones que más convengan para ganar, cumpliendo siempre las reglas del juego y sabiendo que los demás jugadores también influyen en los resultados con sus decisiones.

Los juegos pueden clasificarse en juegos cooperativos y juegos no cooperativos. En los juegos cooperativos se contempla la opción de que los jugadores lleguen a acuerdos sobre qué decisiones debe tomar cada cual. En los no cooperativos el análisis está restringido a las decisiones que tomaría cada jugador en ausencia de acuerdo previo con los demás.

Dentro de los juegos no cooperativos se consideran dos distinciones: juegos estáticos y juegos dinámicos. En los primeros, los jugadores toman sus decisiones simultáneamente y deciden sin saber qué decisión han tomado los demás. En los segundos puede darse que algún jugador conozca ya las decisiones de otro antes de tomar la suya.

Volviendo de nuevo al caso general, se puede hacer otra distinción entre los juegos en función de la información a la que tienen acceso los jugadores. En un juego con información completa todos los jugadores conocen con total certeza las consecuencias, tanto para ellos como para los demás, de todas las decisiones, así como todo lo que ha sucedido en el juego hasta el instante de la decisión. En un juego con información incompleta, algún jugador desconoce parte de la información del juego.

Previamente a definir formalmente los conceptos básicos de la teoría de juegos, cabe describirlos de manera literaria enfatizando qué representan desde el punto de vista intuitivo:

- Jugadores. Son los individuos que toman decisiones con el fin de maximizar su utilidad.
- Utilidad. Es la valoración que da cada jugador a los pagos de los posibles resultados del juego.
- Pagos. Son las consecuencias que se dan para cada jugador dependiendo del resultado del juego.
- Resultados del juego. Son las posibles conclusiones del juego, llevando asociada cada una de ellas unas consecuencias para cada jugador.
- Perfil de estrategias. Es un conjunto de estrategias en el que cada una de ellas está asociada a un jugador.
- Estrategia. Es el plan completo de acciones con las que un jugador puede proponerse jugar a un juego.
- Acciones. Son las decisiones que puede tomar cada jugador en cada momento en que le toque jugar.

A la hora de representar los juegos, se suele hacer fundamentalmente de dos maneras diferentes, en función de en qué se quiera hacer más énfasis. Estas son la forma estratégica o forma normal y la forma extensiva. En ambas se especifican los jugadores, las acciones y los pagos, organizándolos tabularmente en la forma estratégica y en diagrama de árbol en la forma extensiva.

Definición 1.1. Un juego en forma extensiva Γ está caracterizado por el conjunto

$$\Gamma = \left\{ J, (X,\xi), (A,\alpha), \{X_i\}_{i\in J}, \{H_i\}_{i\in J}, \{A_h\}_{\substack{h\in H_i\\ i\in J}}, \{p_h\}_{h\in H_0}, r \right\}, \qquad (1.1)$$

donde:

1. $J = \{0,1,2,\ldots,n\}$ es el conjunto de jugadores o agentes, representando el jugador 0 (o J0) los movimientos del azar.
2. X es el conjunto de nodos, es decir, el conjunto de las posibles situaciones del juego.

   La función $\xi : X \longrightarrow X$ asocia, a cada nodo x, su nodo inmediatamente predecesor, excepto para el nodo origen o, que es el inicio del juego, para el cual $\xi(o) = o$.

3. A es el conjunto de todas las acciones posibles de todos los jugadores.

   La función $\alpha : X \setminus \{o\} \longrightarrow A$ asocia, a cada nodo distinto del origen, la acción $\alpha(x)$ que lleva $\xi(x)$ a x. Se verifica que si $x' \neq x''$ con $\xi(x') = \xi(x'') = x$, entonces $\alpha(x') \neq \alpha(x'')$.

4. X_i es el conjunto de nodos en los cuales el jugador $i \in J$ ha de elegir una acción (nodos de decisión del jugador i).

   Nótese que un jugador $i \in J$ puede tener que realizar una acción sin saber exactamente en qué nodo se encuentra (por ejemplo, en juegos con información incompleta). Por tanto $X_i = \bigcup_{\mu \in I_i} X_i^\mu$ donde I_i indexa la partición definida por la relación xR_ix' si y solo si

   - $x, x' \in X_i$
   - El jugador i no distingue entre x y x'.

5. Para cada $i \in J$ se define $H_i(x), x \in X_i$ como $H_i(x) = X_i^x$, es decir, la función que asigna, a cada nodo $x \in X_i$ el conjunto de nodos de decisión de i indistinguibles de x para el jugador i.

   Si $x \in X_i$, $H_i(x)$ se denomina conjunto de información del jugador i (para el nodo x).

   Con un pequeño abuso de notación $H_i = \{H_i(x), x \in X_i\}$ es la familia de conjuntos de información del jugador i y $H = \{H_i\}_{i \in I}$.

6. Dado $h \in H_i$ un conjunto de información, A_h es el conjunto de acciones disponibles en todos los nodos de h.

7. Para cada $h \in H_0, p_h : A_h \longrightarrow [0, 1]$ es una función que asocia una probabilidad a las acciones debidas al azar (jugador 0) en cada conjunto de información del mismo.

8. Sea $s(x) = \{y \in X : \xi(y) = x\}$, el conjunto de nodos inmediatamente siguientes a x. Se define $T(X) = \{x \in X : s(x) = \varnothing\}$ como el conjunto de nodos terminales y $r : T(X) \longrightarrow \mathbb{R}^n$ es la función que asocia cada nodo terminal a un vector de pagos. En dicho vector de pagos la coordenada i-ésima se corresponde con el pago (o utilidad) asociado al jugador i.

A mayores, de nuevo con un pequeño abuso de notación, se define el conjunto de las acciones disponibles del jugador i a partir de $x \in X_i$ por

$$A(x) = \{a \in A_{H_i(x)} : \exists x' \in s(x) \text{ con } a = \alpha(x')\}. \tag{1.2}$$

Nótese que si x y x' pertenecen al mismo conjunto de información, necesariamente $A(x) = A(x')$, ya que, en caso contrario, el jugador tendría un modo de distinguir x y x'.

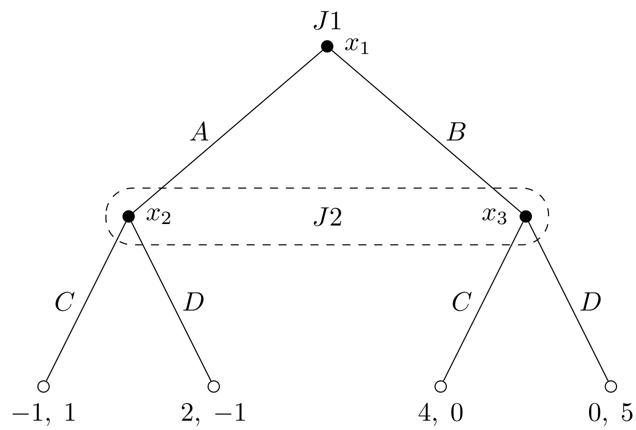

Figura 1.1: Juego con dos jugadores y dos conjuntos de información, el del segundo jugador se representa por el rectángulo punteado.

En la Figura 1.1, aparecen los pagos de cada jugador en los nodos terminales. El primer número se corresponde con el pago de J1 así como el segundo es el correspondiente a J2. Ambos jugadores tienen dos acciones posibles en sus respectivos conjuntos de información.

La siguiente Definición 1.2 formaliza el concepto de estrategia, que intuitivamente interpretamos como el plan de acción de un jugador. Las estrategias han de considerarse para cada conjunto de información del juego pues son los estadios en los cuales el jugador puede tomar una decisión.

Definición 1.2. Una estrategia pura s_i para el jugador $i \in \{1, \dots, n\}$ es una función

$$\begin{array}{rccc} s_i : & H_i & \longrightarrow & A \\ & h & \longmapsto & s_i(h) \end{array}, \tag{1.3}$$

con $s_i(h) \in A_h$. Se denota por S_i al conjunto de todas las estrategias puras posibles de i.

Definición 1.3. Dada una estrategia pura $s_i \in S_i$ para cada jugador $i = 1, 2, \dots, n$ se dice que

$$s = (s_1, s_2, \dots, s_n) \in S_1 \times S_2 \times \cdots \times S_n = S, \tag{1.4}$$

es un perfil de estrategias puras.

Nótese que un perfil de estrategias determina un desarrollo completo del juego y conduce irremediablemente a un nodo terminal, salvo que existan movimientos debidos al azar.

Definición 1.4. Dado el perfil de estrategias puras s, se denota $n(s)$ al nodo terminal que se alcanza tras el desarrollo completo del juego y $u_i(s) = r_i(n(s))$ al pago que recibe el jugador i.

Definición 1.5. Un juego en forma estratégica (o en forma normal) G está caracterizado por el conjunto

$$G = \{J, (S_i)_{i\in J}, (u_i)_{i\in J}\}, \tag{1.5}$$

donde S_i, $u_i, i \in I$ vienen dados en la Definición 1.2 y la Definición 1.4 respectivamente.

Nota. *Se puede prescindir de J en (1.5), ya que está implícito en el conjunto de índices de S_i y u_i.*

Nótese que en la Definición 1.5, u_i representa la función

$$\begin{array}{rccc} u_i: & S & \longrightarrow & \mathbb{R} \\ & s & \longmapsto & r_i(n(s)) \end{array}. \tag{1.6}$$

En caso de estar considerando un juego que contenga movimientos de azar, las funciones u_i se sustituyen por las funciones de utilidad de Von Neumann-Morgenstern asociadas a los resultados del juego. Dicha función es el pago esperado de cada jugador en sentido probabilístico (ver [3]).

La Tabla 1.1 da un ejemplo de un juego en forma estratégica. Las filas y columnas representan las posibles estrategias de cada jugador mientras que los números hacen referencia a los pagos de cada perfil de estrategias para J1 y J2 respectivamente.

G		J2		
		α	β	γ
J1	a	2, 2	0, 2	0, 0
	b	0, 2	2, 0	0, 1
	c	0, 0	1, 0	1, 1

Tabla 1.1: Juego con dos jugadores y tres posibles estrategias por jugador.

Definición 1.6. Se dice que una información I es de dominio público, lo que se denotará por $p(0)$, si para el conjunto de jugadores J todos los jugadores conocen I y se cumplen las proposiciones, definidas de manera inductiva,

- $p(n) =$ "Todos los jugadores de J conocen $p(n-1)$", con $n = 1, 2, \ldots$.

Así pues, de ahora en adelante se considera de dominio público el conocimiento de la estructura completa del juego por parte de todos los jugadores. Es decir, todos los jugadores conocen la estructura del juego, todos los jugadores conocen que todos los jugadores conocen la estructura del juego...

Dado que el juego que utilizaremos como modelo posteriormente es el dilema del prisionero, los resultados descritos a continuación se corresponden a juegos estáticos con información completa. Es decir, juegos en los cuales los jugadores toman las decisiones de manera simultánea y conocen con total certeza los actos llevados a cabo hasta el momento, así como las consecuencias de sus decisiones.

Merece la pena indicar que, pese a que hasta el momento se han considerado juegos con un conjunto de acciones arbitrario, de ahora en adelante se formalizarán desarrollos sobre juegos finitos, es decir, aquellos tales que $|A| \in \mathbb{N}$ ($\Leftrightarrow |S| \in \mathbb{N}$).

1.2. Soluciones de un juego

Dado que en los juegos intervienen varios jugadores, el concepto de solución no es exactamente igual al utilizado de manera usual. Pese a que cada agente o jugador pueda identificar correctamente los resultados óptimos para sí, estos están condicionados por las elecciones tomadas por el resto de agentes. Además, en el caso general, pueden entrar en conflicto las preferencias de los distintos agentes.

Así pues, se llamará solución de un juego a un conjunto de perfiles de estrategias tal que es razonable pensar que los jugadores tomarán decisiones pertenecientes a dicho conjunto. Por otro lado, se llamará concepto de solución a un procedimiento que permita obtener una solución de manera precisa y argumentada.

1.2.1. Soluciones mediante argumentos de dominación

Definición 1.7. Sean el juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$ y s_i', s_i'' dos estrategias del jugador i. Se dice que s_i' está dominada (o débilmente dominada) por s_i'' si se cumple

$$u_i(s_1, \ldots, s_{i-1}, s_i', s_{i+1}, \ldots, s_n) \leq u_i(s_1, \ldots, s_{i-1}, s_i'', s_{i+1}, \ldots, s_n), \qquad (1.7)$$

para toda combinación de estrategias s_{-i} de los otros jugadores y para alguna de las posibles combinaciones se cumple de modo estricto.

Nota. *Para el resto del TFG se utilizará la notación* $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$.

En las condiciones de la Definición 1.7 es correcto decir que "s_i'' domina a s_i'" o, equivalentemente, "s_i' está dominada por s_i''".

Definición 1.8. Sean el juego $G = \{S_1, \dots, S_n; u_1, \dots, u_n\}$ y s_i', s_i'' dos estrategias del jugador i. Se dice que s_i' está estrictamente dominada por s_i'' si se cumple

$$u_i(s_1, \dots, s_{i-1}, s_i', s_{i+1}, \dots, s_n) < u_i(s_1, \dots, s_{i-1}, s_i'', s_{i+1}, \dots, s_n), \tag{1.8}$$

para toda combinación de estrategias s_{-i} de los otros jugadores.

Si una estrategia s_i'' domina estrictamente a otra estrategia s_i' significa que, sean cuales sean las estrategias llevadas a cabo por el resto de jugadores, al jugador i le conviene adoptar s_i'' frente a s_i' siempre. Por tanto, un jugador racional nunca jugará estrategias dominadas y mucho menos estrategias estrictamente dominadas.

Definición 1.9. Sean el juego $G = \{S_1, \dots, S_n; u_1, \dots, u_n\}$ y s_i' una estrategia del jugador i.

- Se dice que s_i' es no dominada si no existe ninguna estrategia del jugador i que la domine.
- Se dice que s_i' es no dominada estrictamente si no existe ninguna estrategia del jugador i que la domine estrictamente.

Si bien la definición anterior describe estrategias maximales bajo la relación de orden de la dominación, cabe dar también una descripción de los máximos bajo dicha relación.

Definición 1.10. Sean el juego $G = \{S_1, \dots, S_n; u_1, \dots, u_n\}$ y s_i' una estrategia del jugador i.

- Se dice que s_i' es dominante cuando se cumple

$$u_i(s_1, \dots, s_{i-1}, s_i, s_{i+1}, \dots, s_n) \leq u_i(s_1, \dots, s_{i-1}, s_i', s_{i+1}, \dots, s_n), \tag{1.9}$$

  para toda estrategia s_i de i y toda combinación de estrategias s_{-i} del resto de jugadores.
- Se dice que s_i' es estrictamente dominante si la premisa anterior se satisface estrictamente para toda estrategia $s_i \neq s_i'$.

Un jugador puede tener varias estrategias dominantes, pues no se ha impuesto en la definición que la ecuación (1.9) se satisfaga estrictamente al menos en algún caso.

Estamos ahora en condiciones de introducir el primer concepto de solución: el Uso de Estrategias Dominantes (UED).

Definición 1.11. Dado un juego G, pertenecen a su solución UED ($\mathrm{S}^{\mathrm{UED}}$) todos aquellos perfiles de estrategias en los que cada jugador asume una estrategia dominante.

Que cada jugador tenga al menos una estrategia dominante es una condición muy restrictiva para la existencia de soluciones, pues en la inmensa mayoría de juegos esto no ocurre. Esto lleva a buscar un concepto de solución basado en argumentos de dominación más aplicables.

Definición 1.12. Dado un juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, llamamos Eliminación Iterativa Estricta (EIE) al algoritmo iterativo consistente en la repetición del siguiente proceso:

- Se eliminan a la vez, y de cada uno de los jugadores, todas las estrategias que estén estrictamente dominadas en el juego G_m ($G_0 = G$), obteniéndose el juego reducido G_{m+1}.

El algoritmo concluye cuando no quedan estrategias que eliminar para ningún jugador. El conjunto de las estrategias supervivientes para cada jugador i se denota por $\hat{\Omega}_i$.

Definición 1.13. Dado un juego G, pertenecen a su solución EIE ($\mathrm{S}^{\mathrm{EIE}}$) todos aquellos perfiles de estrategias en los que cada jugador i asume una estrategia perteneciente a $\hat{\Omega}_i$.

Dado que la existencia de estrategias estrictamente dominadas sigue siendo una característica muy poco común en los juegos, en general se utiliza la versión débil del concepto de solución anterior.

Definición 1.14. Dado un juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, llamamos Eliminación Iterativa Débil (EID) al algoritmo iterativo dado en la Definición 1.12, considerando estrategias débilmente dominadas en lugar de estrictamente dominadas. El conjunto de estrategias supervivientes se denota por Ω_i.

Definición 1.15. Dado un juego G, pertenecen a su solución EID ($\mathrm{S}^{\mathrm{EID}}$) todos aquellos perfiles de estrategias en los que cada jugador i asume una estrategia perteneciente a Ω_i.

Definición 1.16. Dado un juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, se dice que G es resoluble por dominación si, tras aplicar el proceso EID, para todo jugador i y toda combinación de estrategias $s_{-i} \in \Omega_1 \times \cdots \times \Omega_{i-1} \times \Omega_{i+1} \times \Omega_n$ se cumple:

$$\text{Si } \; s_i, s_i' \in \Omega_i \; \text{ entonces } \; u_i(s_i, s_{-i}) = u_i(s_i', s_{-i}). \tag{1.10}$$

Definición 1.17. Sea G un juego resoluble por dominación. Cada perfil de estrategias $s \in \Omega_1 \times \cdots \times \Omega_n$ se denomina equilibrio sofisticado.

Las soluciones descritas en este epígrafe se han propuesto suponiendo, primero, que los jugadores son racionales (Definición 1.11), posteriormente, considerando que también conocen la racionalidad del resto de jugadores (Definiciones 1.13 y 1.15). Así se obtienen conceptos de solución que reducen el número de casos a considerar cuando se busca una estrategia óptima para cada jugador. Por el contrario, en el epígrafe siguiente, se buscan las propiedades que ha de tener un perfil de estrategias para constituir una solución del juego.

1.2.2. Soluciones mediante argumentos de equilibrio

Definición 1.18. Dado el juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, se dice que el perfil de estrategias puras $(s_i^*, \ldots, s_n^*)$ es un Equilibrio de Nash (EN) si para cada jugador i se cumple

$$u_i(s_1^*, \ldots, s_{i-1}^*, s_i^*, s_{i+1}^*, \ldots, s_n^*) \geq u_i(s_1^*, \ldots, s_{i-1}^*, s_i, s_{i+1}^*, \ldots, s_n^*), \tag{1.11}$$

para toda $s_i \in S_i$.

Nota. *Por analogía con la notación* $\mathrm{S}^{\mathrm{EID}}$*, se denomina* S^{EN} *al conjunto de perfiles que son EN de un juego.*

Es decir, en un EN, para cada jugador i, s_i^* es una respuesta óptima a s_{-i}^*. Luego en dicho equilibrio, a ningún jugador le conviene desviarse de manera unilateral del perfil de estrategias dado.

Cabe destacar que en un EN cada jugador no tiene por qué alcanzar necesariamente el mejor pago posible. Es más, ni siquiera significa que alguno de los jugadores lo alcance. Hilando aún más fino, pueden existir perfiles de estrategias que, para todos y cada uno de los jugadores, conduzcan a pagos superiores a los alcanzables mediante los EN del juego.

De esta forma, la palabra equilibrio evoca precisamente la noción de equilibrio empleada en física (mínimo relativo) más que a la noción de solución. Al fin y al cabo se puede interpretar el perfil de estrategias como un sistema mecánico y la función de pagos de cada jugador con signo negativo como un potencial. Entonces en un EN, cualquier pequeña perturbación por parte de un jugador conduce a un estado del sistema con mayor potencial (menor valor absoluto en su función de utilidad) y, en consecuencia, la evolución natural del sistema conduce de nuevo al equilibrio (la estrategia natural de cada jugador es no desviarse del EN).

Precisamente la interpretación anterior invita a pensar que cualquier solución del juego ha de ser EN pues los equilibrios de Nash son máximos relativos de la función de utilidad de cada jugador. Entiéndase que se asume que la utilidad es función únicamente de la estrategia del jugador al que hace referencia: $u_i(s_1^*, \ldots, s_i, \ldots, s_n^*) = f(s_i)$.

Proposición 1.19. Sea $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$ un juego y sea un perfil de estrategias $s^* = (s_1^*, \ldots, s_i^*, \ldots, s_n^*)$. Entonces, si s^* está constituido por estrategias dominantes, es EN.

Dem. Si las estrategias de s^* son dominantes, cualquier estrategia s_i^* de ellas es respuesta óptima para i a cualquier combinación de estrategias de los otros jugadores, en particular lo es a s_{-i}^*, luego es EN. □

Como consecuencia de la Proposición 1.19, los elementos de $\mathrm{S}^{\mathrm{UED}}$ se denominan equilibrios en estrategias dominantes.

Cabría pensar que en las condiciones de la proposición anterior s^* es el único EN del juego, así como también cabría pensar que todos los EN podrían estar en $\mathrm{S}^{\mathrm{UED}}$. El juego contraejemplo (CE) (Tabla 1.2) da una respuesta negativa a ambas afirmaciones simultáneamente.

CE		J2	
		α	β
J1	a	**2**, **1**	1, **0**
	b	**1**, 1	1, 1

Tabla 1.2: Juego CE.

En CE las estrategias a y α son dominantes para J1 y J2 respectivamente, pero hay dos EN: (a, α) y (b, β). Cabe observar que, además, b y β son estrategias dominadas, luego $(b, \beta) \notin \mathrm{S}^{\mathrm{UED}}$.

Dado que estos son los primeros EN que vemos en la práctica, merece la pena explicar al menos uno de forma más detallada (los pagos a los que se hace alusión aparecen en negrita en la definición del juego). Considerando el equilibrio (a, α), J1 obtiene un pago de 2 unidades Si decidiera desviarse optando por la estrategia b, su pago sería $u_1(b, \alpha) = 1$. Por parte de J2, su pago es 1, mientras que si se desviara del equilibrio, optando por la estrategia β, obtendría $u_2(a, \beta) = 0$. Luego, para ambos jugadores, desviarse del equilibrio unilateralmente conlleva una menor utilidad.

Proposición 1.20. Sea $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$ un juego y sea un perfil de estrategias $s^* = (s_1^*, \ldots, s_i^*, \ldots, s_n^*)$. Entonces:

1. Si s^* es EN, entonces $s^* \in \mathrm{S}^{\mathrm{EIE}}$.

2. Si s^* es el único elemento de S^{EIE}, entonces s^* es el único EN de G.

Dem.

1. Se procederá por reducción al absurdo, partiendo de que s^* es EN. Supongamos que s^* no pertenece a $\mathrm{S}^{\mathrm{EIE}}$. Entonces, para un cierto jugador i, la estrategia s_i^* está estrictamente dominada por otra estrategia s_i'. Si esto fuera así, la estrategia s_i^* no sería óptima para la combinación de estrategias s_{-i}^*, pues la estrategia s_i' reportaría un pago estrictamente mayor. Por tanto, se llega a un absurdo pues s^* no podría ser EN.

2. Recíprocamente, sea s^* es el único elemento de $\mathrm{S}^{\mathrm{EIE}}$. Supongamos que no es EN y que, por tanto, existe al menos una estrategia s_i^* que no es respuesta óptima a la combinación de estrategias s_{-i}^*. Sea s_i' una estrategia de i, distinta de s_i^*, que sí es respuesta óptima a s_{-i}^*. La estrategia s_i' ha debido ser eliminada en alguna etapa del algoritmo EIE, pues s_i^* es la única estrategia superviviente de i. Esto es imposible, pues considerando las estrategias s_{-i}^* del resto de jugadores, ninguna otra estrategia reporta un pago mayor para i que s_i' y, por tanto, ninguna la domina.

□

El juego E (Tabla 1.3) tiene tres EN y el algoritmo EID elimina únicamente las estrategias c y γ. En él se observa que hay un EN que no es solución EID, (c, γ). Hay dos EN que son solución EID, (a, α) y (b, β), y dos soluciones EID que no son EN, (a, β) y (b, α).

E		J2		
		α	β	γ
J1	a	2, 1	0, 0	1, 0
	b	1, 0	3, 3	0, 0
	c	1, 1	0, 0	1, 1

Tabla 1.3: Juego E.

Por otro lado, todas las soluciones EID son soluciones EIE, pues el proceso EID es menos severo eliminando estrategias; en cambio, no necesariamente una solución EIE ha de ser EID. Así pues, el diagrama de Venn de la Figura 1.2 da una idea de las relaciones de contención de las distintas nociones de solución dadas hasta ahora. Aún así, este diagrama puede variar dependiendo del juego al "menguar" alguno de los conjuntos.

A título informativo, es curioso el comportamiento de las soluciones UED en cuanto a contenciones, pues si existe sólo una, entonces $\mathrm{S}^{\mathrm{UED}} = \mathrm{S}^{\mathrm{EID}}$, pero si existe más de una, entonces $\mathrm{S}^{\mathrm{EID}} = \varnothing$. Por otra parte siempre se tiene que $\mathrm{S}^{\mathrm{UED}} \subset \mathrm{S}^{\mathrm{EN}} \subset \mathrm{S}^{\mathrm{EIE}}$.

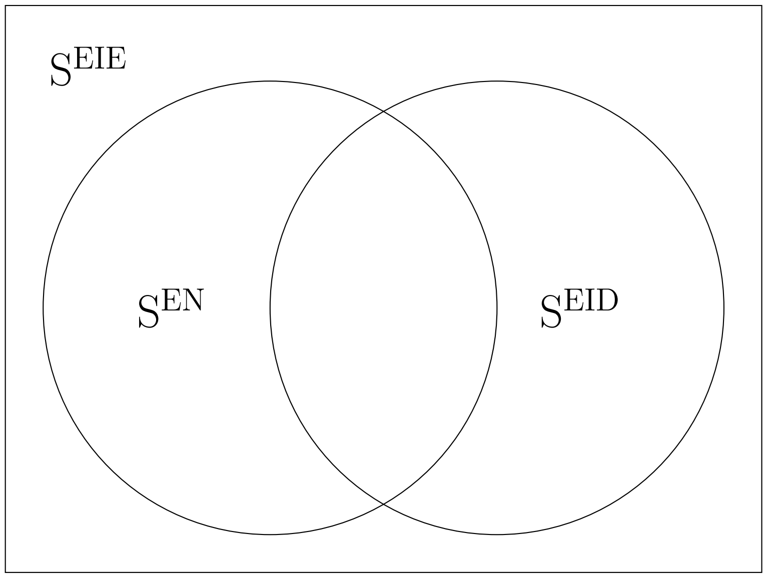

Figura 1.2: Contenciones generales de las distintas nociones de solución.

Anteriormente se ha indicado que los EN de un juego no tienen que corresponderse con el máximo de utilidad para ningún jugador. Las dos definiciones siguientes introducen los conceptos de eficiencia y óptimo de Pareto, que responden a una búsqueda intuitiva del beneficio máximo común.

Definición 1.21. Dado el juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, se dice que el perfil de estrategias $s = (s_1, \ldots, s_n)$ está dominado en el sentido de Pareto por el perfil $s' = (s'_1, \ldots, s'_i, \ldots, s'_n)$ si, y solo si, la desigualdad $u_i(s'_1, \ldots, s'_n) \geq u_i(s_1, \ldots, s_n)$ se cumple para todo jugador i y, además, para alguno de ellos se satisface la desigualdad estricta.

Definición 1.22. Dado el juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, se dice que el perfil de estrategias $s = (s_1, \ldots, s_n)$ es óptimo de Pareto (es eficiente en sentido de Pareto) si, y solo si, no está dominado en el sentido de Pareto por ningún otro perfil. Recíprocamente, se dice que es ineficiente en el sentido de Pareto si está dominado por algún otro perfil.

Cabe destacar que pueden existir varios óptimos de Pareto, incluso con vectores de pagos distintos. De hecho, al cambiar de un óptimo a otro con distinto vector de pagos, al menos uno de los jugadores sale perdiendo y al menos otro sale ganando.

Es muy importante comprender que el EN representa el perfil cuyas estrategias son apropiadas desde el punto de vista individual, donde cada jugador busca el beneficio propio y considera que el resto de jugadores harán lo mismo. Por el contrario, el óptimo de Pareto indica las estrategias apropiadas desde el punto de vista social, pues es una situación que no puede mejorar a la vez para varios jugadores. Si mejora para un jugador, otro sale perjudicado, luego no hay ninguna otra estrategia que, globalmente, sea mejor que el óptimo de Pareto.

La ventaja del EN sobre las anteriores nociones de solución es que contempla las estrategias adecuadas para un jugador que supone que el resto jugarán intentando maximizar su beneficio y los demás jugadores, sabiendo que él también lo hará. En cualquier caso, conviene señalar que en todas estas definiciones y razonamientos se ha supuesto que las conjeturas hechas por cada jugador han sido correctas para poder llegar a los perfiles de soluciones.

1.2.3. Equilibrios en estrategias mixtas

Definición 1.23. Dado $S_i = \{s_i^1, s_i^2, \ldots, s_i^{k_i}\}$ el conjunto de las k_i estrategias puras del jugador i. Se denomina estrategia mixta del jugador i a toda distribución de probabilidad $\sigma_i = (\sigma_i^1, \sigma_i^2, \ldots, \sigma_i^{k_i})$ sobre S_i. El conjunto de estrategias mixtas de un jugador i se denota por $\Delta(S_i)$.

Se interpreta σ_i como la estrategia consistente en jugar la estrategia pura s_i^j con probabilidad σ_i^j donde $\sigma_i^j \geq 0$ y $\sum_{j=1}^{k_i} \sigma_i^j = 1$ para cada $j, i = 1, 2, \ldots, n$.

Nótese que el concepto de estrategia mixta generaliza el de estrategia pura, pues basta considerar los vectores canónicos de tamaño k_i para recuperar las estrategias puras del juego. En cualquier caso, cabe reseñar que para definir una estrategia mixta es necesario indicar el conjunto de estrategias puras al que hace referencia. Así pues, de ahora en adelante se entiende que una estrategia es cualquier combinación lineal convexa de las estrategias puras de un juego (estrategia mixta).

Definición 1.24. Dada σ_i una estrategia mixta y $S_i = \{s_i^1, \ldots, s_i^k\}$ el conjunto de estrategias puras del jugador i, se denomina soporte de σ_i al subconjunto de S_i

$$\mathrm{SOP}(\sigma_i) = \{s_i^j \in S_i \mid \sigma_i^j > 0\}. \tag{1.12}$$

Definición 1.25. Las estrategias mixtas σ_i tales que $\mathrm{SOP}(\sigma_i) = S_i$ se denominan completas y aquellas cuyo soporte no es una única estrategia pura se denominan propias.

La utilización de este tipo de estrategias implica que, en la práctica, el desarrollo del juego se convierte en un proceso estocástico. Como consecuencia, la función de pagos de cualquier jugador deja de ser determinista y pasa a ser aleatoria. Así pues, en el caso de estrategias mixtas, las funciones de utilidad de cada jugador se convierten en pagos esperados, representados con mayúsculas, tal y como se definen a continuación.

Definición 1.26. Dado el juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, se dice que el perfil de estrategias mixtas $\sigma^* = (\sigma_i^*, \ldots, \sigma_n^*)$ es un Equilibrio de Nash (EN) si para cada jugador i se cumple

$$U_i(\sigma_1^*, \ldots, \sigma_{i-1}^*, \sigma_i^*, \sigma_{i+1}^*, \ldots, \sigma_n^*) \geq U_i(\sigma_1^*, \ldots, \sigma_{i-1}^*, \sigma_i, \sigma_{i+1}^*, \ldots, \sigma_n^*), \tag{1.13}$$

para toda σ_i de $\Delta(S_i)$.

Nota. *Toda formalización descrita a partir del concepto de EN de la Definición 1.18 puede extenderse sin dificultad a la noción generalizada recién definida. Es decir, de ahora en adelante, se remplaza la Definición 1.18 por la Definición 1.26.*

La consideración de estrategias mixtas en el análisis de un juego permite acceder a equilibrios de Nash anteriormente inexistentes. Los pagos esperados de estos equilibrios, así como de cualquier estrategia mixta, son combinación convexa de los pagos de las estrategias puras de su soporte. En consecuencia están acotados superior e inferiormente por las ganancias máxima y mínima de las estrategias puras del soporte respectivamente. Por tanto, sólo pueden constituir equilibrios de Nash aquellas estrategias cuyos elementos del soporte generen las mismas ganancias dada la combinación de estrategias del resto de jugadores. En caso contrario no sería respuesta óptima pues existiría una estrategia pura que reportaría un pago mayor. Esto se formaliza a continuación.

Teorema 1.27. Dado el juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, el perfil de estrategias mixtas $\sigma^* = (\sigma_i^*, \ldots, \sigma_n^*)$ es un EN si y solo si para cada jugador i con estrategia mixta σ_i^* se tiene que

$$\text{Si } s_i^j \in \mathrm{SOP}(\sigma_i^*) \text{ entonces } s_i^j \text{ es respuesta óptima a } \sigma_{-i}^*. \tag{1.14}$$

Dem. Supongamos cierta la segunda parte de la doble implicación. Sea M_i^* la máxima utilidad alcanzable por i ante σ_{-i}^*. Considerando que la notación (x_{-i}, y_i) equivale a $(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$ se observa que $\forall s_i^j \in \mathrm{SOP}(\sigma_i^*)$ se tiene que:

$$U_i(\sigma_{-i}^*, s_i^j) = \text{máx}\{U_i(\sigma_{-i}^*, s_i), s_i \in S_i\} = M_i^*. \tag{1.15}$$

Por tanto, se puede concluir que $\forall \sigma_i \in \Delta(S_i)$ se tiene que

$$U_i(\sigma_{-i}^*, \sigma_i^*) = \sum_j \sigma_i^{j*} \,\text{máx}\{U_i(\sigma_{-i}^*, s_i^j), s_i \in S_i\} = \sum_j \sigma_i^{j*} M_i^* = M_i^* \geq U_i(\sigma_{-i}^*, \sigma_i). \tag{1.16}$$

Supongamos ahora que σ^* es EN. Considérese para cada jugador i la estrategia pura s_i^j, perteneciente al soporte de σ_i^*. Se demuestra por reducción al absurdo que s_i^j es respuesta óptima a σ_{-i}^*; si no lo fuera, existiría una estrategia pura s_i^m tal que

$$U_i(\sigma_{-i}^*, s_i^m) > U_i(\sigma_{-i}^*, s_i^j). \tag{1.17}$$

Tomemos ahora la estrategia mixta σ_i' idéntica a σ_i^* salvo para el jugador i, que seguirá la estrategia pura s_i^j con probabilidad 0 y la estrategia pura s_i^m con probabilidad $\sigma_i^{j*} + \sigma_i^{m*}$. Entonces, tal y como se ve a continuación, σ_i' es una respuesta estrictamente mejor que σ_i^* a σ_{-i}^*, lo cual es una contradicción.

$$\begin{aligned} U_i(\sigma_{-i}^*, \sigma_i') &= \sum_{h\neq j, h\neq m} \sigma_i^{h*} U_i(\sigma_{-i}^*, s_i^h) + (\sigma_i^{j*} + \sigma_i^{m*})\, U_i(\sigma_{-i}^*, s_i^m) \\ &> \sum_{h\neq j, h\neq m} \sigma_i^{h*} U_i(, \sigma_{-i}^*, s_i^h) + \sigma_i^{j*} U_i(\sigma_{-i}^*, s_i^j) + \sigma_i^{m*} U_i \sigma_{-i}^*, s_i^m) = U_i(\sigma_{-i}^*, \sigma_i^*). \end{aligned} \tag{1.18}$$

□

A continuación se introduce un conjunto de definiciones *ad hoc* empleadas en el Teorema 1.29 y el Teorema 1.30, que sirve a su vez para demostrar el último resultado de la Sección.

Definición 1.28. Sea C un subconjunto de $\mathbb{R}^m$ como espacio euclídeo. Decimos que

1. C es convexo si dados dos puntos cualesquiera $x, y \in C$, se cumple para todo $\lambda \in [0,1]$, $\lambda x + (1-\lambda)y \in C$. Es decir, cualquier combinación convexa de x e y pertenece a C.

2. C es abierto si dado cualquier punto $x \in C$ existe una bola abierta $B(x,r)$ de centro x y radio r contenida en C.

3. C es cerrado si su complementario $\mathbb{R}^m \setminus C$ es abierto.

4. C es acotado si existe una bola $B(0,r)$ para un cierto $r > 0$ tal que $C \subset B(0,r)$.

5. C es compacto si es cerrado y acotado.

6. Dada una función $f : C \to B \subset \mathbb{R}^p$, es continua en C si para cada sucesión $\{x_j\}_{j=1}^{\infty}$ de puntos de C que converge hacia $x \in C$, la sucesión $\{f(x_j)\}_{j=1}^{\infty}$ converge hacia $f(x)$.

7. Dada una correspondencia $g : C \to B \subset \mathbb{R}^p$ con B compacto, g es hemicontinua superiormente en C si para todas las sucesiones $\{x_j\}_{j=1}^{\infty}$ de puntos de C con límite $x \in C$ y todas las sucesiones $\{y_j\}_{j=1}^{\infty}$ de puntos de B con límite $y \in B$ tales que $y_m \in g(x_m)$, se verifica que $y \in g(x)$.

8. Dada una correspondencia $g : C \to B \subset \mathbb{R}$ con C convexo, g es cuasicóncava si
$$g(\lambda x + (1-\lambda)y) \geq \min\{g(x), g(y)\},\ \forall \lambda \in [0,1], \forall x, y \in C. \tag{1.19}$$

Teorema 1.29 (Teorema del punto fijo de Kakutani). Sea $g : A \to A$ una correspondencia con dominio en $A \subset \mathbb{R}^m$, donde A es no vacío, compacto y convexo. Si g es hemicontinua superiormente y $g(x) \subset A$ es convexo y no vacío para todo $x \in A$, entonces existe un punto fijo x^* de g, es decir,

$$\exists x^* \in A : x^* \in g(x^*). \tag{1.20}$$

Se puede encontrar una demostración del Teorema 1.29 en [25]. Dicha demostración no se incluye en este texto por requerir un volumen de resultados previos que no es práctico en relación a los objetivos de este trabajo.

Teorema 1.30. Dado un juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, tal que para todo jugador i se cumple:

1. $S_i \subset \mathbb{R}^m$ es un subconjunto no vacío, compacto y convexo.

2. u_i es continua en todo su dominio $S = S_1 \times \cdots \times S_n$ y es cuasicóncava en la variable s_i.

Entonces existe al menos un EN en estrategias puras de G.

Dem. Sea R_i la correspondencia de respuesta óptima de cada jugador i, es decir, la correspondencia que asocia a cada perfil de estrategias puras $s = (s_i, s_{-i})$ el conjunto $R_i(s)$ de las estrategias puras de i que son respuesta óptima a s_{-i}. Definamos ahora la correspondencia global de respuesta óptima:

$$R(s) = R_1(s) \times \cdots \times R_n(s) = \{t = (t_1, \ldots, t_n) \in S : t_i \in R_i(s), \forall i = 1, 2, \ldots, n\}. \tag{1.21}$$

Veamos que R es hemicontinua superiormente. Nótese que S es no vacío, compacto y convexo, por ser S producto cartesiano de conjuntos que lo son; u_i es continua en S_i para cada i por serlo en S, y, por hipótesis S_i es no vacío y compacto. El teorema de Weierstrass asegura que el problema $\max_{y_i \in S_i} u_i(y_i, s_{-i})$ tiene solución óptima, por lo que $R_i(s)$ y, en consecuencia, $R(s)$ son no vacíos. Queda probar que si las sucesiones $\{s^k\}$, $\{t^k\}$, tienen por límite $s \in S$ y $t \in S$ para cada k, con $s^k \in S, t^k \in R(s^k)$ entonces $t \in R(s)$. Ahora bien:

$$s^k \overset{k\to\infty}{\longrightarrow} s \iff \forall i, s_i^k \overset{k\to\infty}{\longrightarrow} s_i \tag{1.22}$$

$$t^k \overset{k\to\infty}{\longrightarrow} t \iff \forall i, t_i^k \overset{k\to\infty}{\longrightarrow} t_i \tag{1.23}$$

$$t^k \in R(s^k) \Longrightarrow \forall i, t_i^k \in R_i(s^k) \Longrightarrow \forall i, u_i(t_i^k, s_{-i}^k) \geq u_i(z_i, s_{-i}^k), \forall z_i \in S_i \tag{1.24}$$

pero dado que u_i es continua en S_i se verifica que $u_i(t_i, s_{-i}) \geq u_i(z_i, s_{-i}), \forall z_i \in S_i$. Por tanto $t_i \in R_i(s), \forall i$ y, en consecuencia, $t \in R(s)$.

Por otra parte, $R(s)$ es compacto por ser cerrado y estar contenido en S, que es compacto. Además, dado que las u_i son cuasicóncavas en S_i, el conjunto de los maximizadores $R_i(s)$ es convexo y, en consecuencia, $R(s)$ también.

Así pues, la correspondencia $R : S \longrightarrow S$ satisface las hipótesis del Teorema 1.29 y, por tanto, tiene un punto fijo s^*. Esto significa que $s^* \in R(s^*)$, con lo que $s_i^* \in R_i(s^*)$ para todo i. En conclusión, s_i^* es respuesta óptima a s_{-i}^* para cada jugador i y s^* es EN de G. □

Teorema 1.31. En todo juego (finito) $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$ existe al menos un EN en estrategias mixtas.

Dem. Sea $S_i = \{s_i^1, \ldots, s_i^{k_i}\}$ y $S = S_1 \times \cdots \times S_n$. Sea $\Delta(G)$ el juego cuyos jugadores son los mismos que los de G pero donde los conjuntos de estrategias puras de cada jugador i son los símplices $\Delta(S_i)$, y cuyos pagos son los pagos esperados U_i del perfil en estrategias mixtas correspondiente en G.

Entonces el juego $\Delta(G) = \{\Delta(S_1), \ldots, \Delta(S_n); u_1, \ldots, u_n\}$ satisface las hipótesis del Teorema 1.30 pues para cada jugador i,

1. $\Delta(S_i)$ es un subconjunto no vacío, compacto y convexo de R^{k_i}.

2. La función de pagos esperados $U_i(\sigma)$ es continua en todo su dominio y cuasicóncava en la variable σ_i por ser una función afín.

Por tanto se puede concluir que existe un EN en estrategias puras del juego $\Delta(G)$, que es un EN en estrategias mixtas de G. □

Con respecto a las estrategias mixtas cabe destacar que existe también la noción de estrategia mixta dominada, que conduce a los algoritmos de eliminación estricta y débil considerando estrategias mixtas. Cabe mencionar que se pueden definir nuevos equilibrios, por ejemplo utilizando los conceptos anteriores, que refinan el EN consiguiendo conjuntos de soluciones más reducidos.

1.3. Dilema del prisionero

A la hora de implementar la metodología de machine learning sobre autómatas que se describirá posteriormente, se eligió como modelo el Dilema del Prisionero Repetido Infinitamente. En esta Sección se tratará el Dilema del Prisionero (DP), juego a partir del cual se define el anterior. El juego fue ideado por Merrill Flood y Melvin Dresher, pero la primera formalización del DP la dio Albert W. Tucker en 1950 según se indica en [15]. En ese momento era un perfecto ejemplo de la situación de los Estados Unidos y la Unión Soviética en el marco de la guerra fría, aunque posteriormente se ha utilizado tanto en ciencias sociales como en biología, psicología, economía y, por supuesto, en teoría de juegos.

El Dilema del Prisionero es un paradigma dentro de los juegos por su simpleza, aplicabilidad y lo relativamente curioso de su análisis, tal y como se verá en las secciones restantes de este primer capítulo. A continuación se dará una descripción particular con unos pagos que no han de considerarse como constituyentes de una única definición del DP. En realidad, se denomina dilema del prisionero a cualquier juego de dos jugadores con pagos que cumplan la misma relación de orden que los de la Tabla 1.5. Una descripción del dilema sería la siguiente:

Dos delincuentes son atrapados por la policía, ambos culpables de un delito menor y sospechosos de otro más grave. Los reos son separados en dos calabozos de manera que no pueden comunicarse entre ellos y a ambos se les propone la misma disyuntiva: pueden cooperar entre ellos negando que hayan cometido el más grave de los delitos o por el contrario pueden traicionar al otro confesando. Si ambos callan, los policías entienden que sólo son culpables del delito menor y reciben una pena de 2 años de prisión; si sólo uno de ellos traiciona al otro, el traidor sale libre por confesar, mientras el otro recibe una pena de 5 años; por último, si ambos confiesan, reciben penas reducidas de 4 años por cooperación con la justicia.

DP		Preso 2	
		Callar	Confesar
Preso 1	Callar	-2, -2	-5, 0
	Confesar	0, -5	-4, -4

Tabla 1.4: Dilema del prisionero con pagos descriptivos.

Si bien los pagos de la Tabla 1.4 describen claramente los años que pasa en prisión cada uno de los presos, en general se utiliza una descripción del juego en la que el menor de los pagos es nulo. Así pues, se utilizará a partir de ahora el DP descrito por la Tabla 1.5. Por otro lado, de ahora en adelante la palabra cooperar hará referencia a la estrategia callar, entendida como cooperación entre jugadores y no como cooperación con la justicia.

DP		Preso 2	
		Callar	Confesar
Preso 1	Callar	3, 3	0, 5
	Confesar	5, 0	1, 1

Tabla 1.5: Dilema del prisionero con pagos estandarizados.

En virtud de la teoría desarrollada en la Sección 1.2 podemos hacer un rápido análisis de este juego. Dado que *Confesar* es la única estrategia dominante para ambos jugadores, se concluye que el perfil $(Confesar, Confesar) \in \mathrm{S}^{\mathrm{UED}}$ es EN. De hecho, también es solución EID y, obviamente, EIE.

Nótese también que, para cualquier jugador y cualquier estrategia del rival, la respuesta óptima es siempre *Confesar*, por ser una estrategia dominante. Entonces, por el Teorema 1.27, sólo pueden ser EN aquellos perfiles de estrategias cuyos soportes tengan como único elemento la estrategia pura *Confesar*. Es decir, no existen equilibrios en estrategias mixtas propias para el dilema del prisionero.

El dilema del prisionero se utiliza constantemente como modelo, entre otras cosas, porque curiosamente su único EN es el único perfil de estrategias que no es óptimo de Pareto. En un análisis desde un punto de vista social, en el DP la estrategia óptima para cada jugador es traicionar al rival rehuyendo el bien común por miedo a que sea el otro quien le traicione y empeore su situación. Tal situación es de sustancial importancia en el estudio del comportamiento de dos agentes tanto en economía como en psicología y tiene aplicación incluso en comportamientos biológicos.

1.4. Dilema del prisionero finitamente repetido

En esta Sección vamos a avanzar hacia el juego último de este capítulo: el dilema del prisionero repetido infinitamente. Para ello, cabe considerar primero su repetición finita, pues introduce múltiples conceptos que serán necesarios posteriormente, así como permite un análisis del juego que servirá en lo sucesivo.

Los juegos repetidos son un tipo particular de juego dinámico, definidos a partir de la repetición de un juego más simple llamado juego de etapa. El hecho de tener una estructura simplifica su análisis con respecto a la mayoría de juegos dinámicos, pero no significa que baste con considerar una acumulación de juegos sueltos. Las decisiones tomadas en una etapa pueden influir en las siguientes, con lo cual los jugadores tendrán esto en cuenta y actuarán en consecuencia.

Antes de introducir los conceptos necesarios sobre juegos dinámicos es necesario indicar que, en el momento de empezar cada etapa, es de dominio público cuál ha sido el transcurso previo del juego además de toda la información de dominio público que ya considerábamos en el caso estático.

Nota. *Se recomienda revisar la Definición 1.1 antes de continuar.*

Definición 1.32. Dado un juego G con información completa en forma extensiva y un nodo de decisión x de G (ver Definición 1.1), decimos que G' es un subjuego de G con inicio en x si G' está definido sobre los elementos de G y cumple que:

1. Contiene únicamente al nodo x y a todos los nodos y tales que $\xi^m(y) = x$ para algún m natural.

2. El nodo x constituye un conjunto de información unitario.

3. G' no altera ningún conjunto de información, es decir, si $y \in X_i^{G'}$ entonces $H_i^{G'}(y) = X_i^{y\,G}$ para todo i.

Los subjuegos de G que no son idénticos a G se denominan subjuegos propios.

Definición 1.33. Sea G un juego en forma extensiva y sea σ un perfil de estrategias de G que es EN. Decimos que σ es un Equilibrio de Nash Perfecto en Subjuegos (ENPS) de G si la restricción de σ a culaquier subjuego de G es un EN en dicho subjuego.

Definición 1.34. Sea G un juego en forma extensiva y sea r un resultado posible de G. Decimos que r es un Resultado Perfecto en Subjuegos (RPS) de G si r puede obtenerse como realización de un perfil σ que es ENPS.

La Definición 1.33 refina el concepto de EN en juegos dinámicos. El EN de un juego dinámico se puede hallar considerando el juego en forma normal, una vez analizadas todas las estrategias posibles de cada jugador. El problema de lo anterior es que se pierde información al considerar la forma estratégica, pues pese a que en general un perfil de estrategias pueda dar lugar a un EN, eso no significa que en cada nodo de decisión vaya a ser razonable seguir la estrategia fijada. La noción de ENPS soluciona el problema anterior, pues en cada nodo de decisión el perfil ENPS da la estrategia óptima para cada jugador.

En cualquier caso, el concepto de ENPS también tiene sus debilidades, pues de manera similar a los EN, todos los jugadores han de coincidir en la predicción de que el resto jugará el perfil que constituye el ENPS. Esto implica que, por ejemplo, en caso de existir varios EN en el juego de etapa, dado un ENPS, todos los jugadores han de predecir en cada nodo de decisión que el resto jugará la estrategia correspondiente al EN que se ha considerado en el ENPS para la etapa en la que se encuentren.

Tal y como se verá a continuación, la existencia de al menos un ENPS está garantizada bajo condiciones bastante generales sobre un juego G.

Definición 1.35. Dado un juego G finito en forma extensiva con información completa y perfecta, llamamos algoritmo de inducción hacia atrás al algoritmo iterativo consistente en la repetición del siguiente proceso:

- Se eliminan todos los subjuegos de G_m ($G_0 = G$) con inicio en x tal que $s(x) \subset T(X_{G_m})$. Estos subjuegos tienen un único jugador y su EN es la acción óptima de dicho jugador, luego se considera que cada x pasa a ser un nodo terminal del juego G_{m+1} con los pagos correspondientes al citado EN.

El algoritmo concluye cuando se llega al nodo inicial de G.

En la definición anterior, la existencia de un EN en cada subjuego viene dada por el Teorema 1.31, ya que cada G_m es finito por serlo G. En caso de existir varios EN para un subjuego basta con tomar uno cualquiera de ellos en el algoritmo.

Teorema 1.36. Todo juego finito con información completa y perfecta presenta al menos un ENPS. Además, si ningún jugador tiene más de una acción óptima en cada nodo de decisión, tal ENPS es único.

Dem. Antes de nada considérese que un nodo x de G_m tiene lejanía l si, en el subjuego de G_m iniciado en x, el máximo número de nodos recorridos para llegar, sin repetir ninguno, desde x a un nodo terminal es l. Sea L la máxima lejanía entre los nodos de decisión de un juego.

En esta demostración G_m hace referencia al subjuego relativo a la m-ésima iteración del algoritmo de inducción hacia atrás.

Veamos por reducción al absurdo que el perfil de estrategias $\sigma = (\sigma_1, \ldots, \sigma_n)$, determinado eligiendo las acciones óptimas en cada iteración del algoritmo de inducción hacia atrás, es EN de G_m para todo m.

Así pues, si no fuera EN, existiría un jugador i para el cual existiría una estrategia $\bar{\sigma}_i$ tal que

$$u_i(\bar{\sigma}_i, \sigma_{-i}) > u_i(\sigma). \tag{1.25}$$

Veamos por inducción en l que lo anterior lleva a contradicción. Nótese que L es natural por ser G_m finito.

Sea $\bar{\sigma}_i(l)$ la estrategia del jugador i dada por:

- $\bar{\sigma}_i(l) = \sigma_i$ en todos los nodos con distancia menor o igual que l.
- $\bar{\sigma}_i(l) = \bar{\sigma}_i$ en todos los nodos con distancia mayor que l.

Entonces, por construcción de σ a través del procedimiento de inducción hacia atrás, se verifica que $u_i(\bar{\sigma}_i(0), \sigma_{-i}) \geq u_i(\bar{\sigma}_i, \sigma_{-i})$. Esto se debe a que el único cambio de $\bar{\sigma}_i(0)$ con respecto a $\bar{\sigma}_i$ es el nodo terminal, donde se elige la decisión óptima (EN del subjuego).

Considérese ahora la hipótesis de inducción $u_i(\bar{\sigma}_i(l-1), \sigma_{-i}) \geq u_i(\bar{\sigma}_i, \sigma_{-i})$. La única diferencia entre $\bar{\sigma}_i(l-1)$ y $\bar{\sigma}_i(l)$ está en las elecciones de i en los nodos con lejanía l. Pero, de nuevo por construcción de σ, las acciones de i relativas a dicho perfil en los nodos de lejanía l son óptimas siempre que se siga el perfil σ en aquellos nodos más próximos a los nodos terminales, lo cual se cumple tanto en $\bar{\sigma}_i(l-1)$ como en $\bar{\sigma}_i(l)$. Entonces

$$u_i(\bar{\sigma}_i(l), \sigma_{-i}) \geq u_i(\bar{\sigma}_i(l-1), \sigma_{-i}), \tag{1.26}$$

y por tanto

$$u_i(\bar{\sigma}_i(L), \sigma_{-i}) \geq u_i(\bar{\sigma}_i, \sigma_{-i}), \tag{1.27}$$

pero $\bar{\sigma}_i(L) = \sigma_i$, que contradice la ecuación (1.25). Por tanto σ es EN de G_m para todo m, en particular también para $G_0 = G$ y se concluye que es ENPS.

Por otro lado, si cada jugador tiene una única acción óptima en cada nodo de decisión, el algoritmo de inducción hacia atrás determina un único ENPS. □

Nota. *La condición de que G sea un juego con información completa es necesaria para la definición de ENPS, con lo que las únicas restricciones efectivas son que sea finito y que tenga información perfecta.*

Definición 1.37. Dado un jugador i, se denota por δ_i al factor de descuento de i tal que para dicho jugador es indiferente

- Cobrar $\delta_i C$ en un instante dado.
- Cobrar C en el siguiente instante.

El factor de descuento suele ser menor o igual que 1, pues los agentes económicos suelen preferir cobrar menos en lugar de tener que esperar para recibir una cantidad mayor tras un cierto tiempo, debido el riesgo que esto conlleva.

Definición 1.38. Sea $\{q_m\}_{m=1,2,\ldots}$ una secuencia de pagos finita o infinita y δ un factor de descuento.

- El valor presente descontado de la secuencia de pagos se define como

$$\mathrm{VP}[\{q_m\}_{m=1,2,\ldots}, \delta] = q_1 + q_2\delta + \ldots q_m\delta^{m-1} + \cdots = \sum_m q_m\delta^{m-1}. \tag{1.28}$$

- El pago medio de la sucesión de pagos es el pago fijo q^* tal que la secuencia constante $\{q^*_m\}_{m=1,2,\ldots}$ donde $q^*_m = q^*, \forall t$ tiene el mismo valor presente descontado que la sucesión $\{q_m\}_{m=1,2,\ldots}$. Es decir, $q^* \sum \delta^{m-1} = \sum q_m \delta^{m-1}$.

Nótese que ahora, el hecho de que el factor de descuento sea menor que 1, garantiza que el valor presente descontado sea finito si la secuencia de pagos está acotada, aunque sea infinita. Asimismo garantiza que el pago medio esté bien definido.

Definición 1.39. Dado un juego $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, decimos que el vector de pagos $(x_1, \ldots, x_n)$ es factible o alcanzable si es combinación convexa de los vectores de pagos correspondientes a perfiles de estrategias puras de G.

Nótese que en la definición anterior, los coeficientes de cada pago correspondiente a un perfil de estrategias puras son iguales para todos los jugadores, lo cual no ocurre necesariamente cuando se consideran los pagos esperados de una estrategia mixta. Por tanto, cualquier vector de pagos correspondiente a un perfil de estrategias mixtas es factible, sin embargo el recíproco no es necesariamente cierto.

Previamente a la definición de juego repetido cabe indicar que a la hora de referirse al juego de etapa se hará llamando acciones a sus estrategias puras, denotadas por a_i, α_i para las estrategias mixtas y g_i para las utilidades de cada jugador i. Se reserva así la notación usual para el juego repetido.

Definición 1.40. Dado un juego $G = \{A_1, \ldots, A_n; g_1, \ldots, g_n\}$ llamado juego de etapa o juego constituyente y un vector de factores de descuento $\delta = (\delta_1, \ldots, \delta_n)$, llamamos juego repetido finitamente $G^T(\delta)$ al juego en el que es de dominio público

1. Antes de comenzar a jugar que:
    i) El factor de descuento de cada jugador i es δ_i.
    ii) Que G se va a jugar T veces.
    iii) Que para cada jugador i los pagos $G^T(\delta)$ son el valor presente para δ_i de la secuencia finita de sus pagos de etapa.
2. Antes de que empiece cualquier etapa, todas las jugadas realizadas en las etapas anteriores.

En lo sucesivo, en aras de la simplificación, se considerará que todos los jugadores comparten el mismo factor de descuento δ y no se considerará la influencia que puede tener sobre cada agente la incertidumbre de que no se pueda producir un pago futuro. El factor de descuento se considera únicamente consecuencia de la preferencia de cada agente por tener liquidez y no se ve influido por la probabilidad de que un pago se haga efectivo o no en el futuro.

Así como $G^1(\delta)$ es precisamente el juego de etapa G, se denota $G^t(\delta)$ a las $t \leq T$ primeras repeticiones del juego repetido $G^T(\delta)$. Se simplifica también la notación cuando $\delta = 1$, denotando al juego repetido por G^T.

Definición 1.41. Dados el juego de etapa $G = \{A_1, \ldots, A_n; g_1; \ldots, g_n\}$ y el juego repetido $G^T(\delta)$, una t-historia es el conjunto

$$h_t = \{(a_1^m, \ldots, a_n^m)\}_{m=1,2,\ldots,t-1}, \tag{1.29}$$

donde $a_i^m \in A_i$ es la acción realizada por el jugador i en la etapa m.

Es decir, una t-historia es el conjunto que recoge todas las acciones tomadas por los distintos jugadores antes de la etapa t.

El conjunto de todas las t-historias h_t es

$$H_t = \left\{\{(a_1^m, \ldots, a_n^m)\}_{m=1,2,\ldots,t-1} : a_i^m \in A_i\right\}. \tag{1.30}$$

Definición 1.42. Dados el juego de etapa $G = \{A_1, \ldots, A_n; g_1; \ldots, g_n\}$ y el juego repetido $G^T(\delta)$, una estrategia σ_i en $G^T(\delta)$ es un plan que determina qué distribución probabilística aplicará i sobre las acciones del juego en cada etapa para cada posible historia hasta ese momento. Es decir, es la T-upla $\sigma_i = \left(a_i^1(h), \ldots, a_i^T(h)\right)$ dada por las T aplicaciones a_i^t

$$\begin{array}{rccc} a_i^t : & H_t & \longrightarrow & \Delta(A_i) \\ & h & \longmapsto & (\alpha_i^1, \ldots, \alpha_i^k) \end{array}. \tag{1.31}$$

Se dice que una estrategia es incondicionada o independiente de la historia cuando para todo $t \leq T$, dadas dos historias cualesquiera $h, h' \in H_t$, se satisface $a_i^t(h) = a_i^t(h')$.

Nota. *No confundir las funciones $a_i^t(h)$ que determinan una cierta estrategia con las acciones a_i^m de una historia dada de i.*

La noción de perfil de estrategias se define de manera análoga al caso estático. En el caso repetido también determina, considerando las acciones de azar correspondientes, un desarrollo completo del juego así como los pagos de etapa correspondientes para cada jugador.

Los subjuegos de $G^T(\delta)$ tienen su inicio al comienzo de cada etapa y son de la forma $G^{T-t}(\delta)$ con $0 \leq t < T$. En particular, existen $|H_t|$ juegos que comienzan en la etapa t.

Teorema 1.43. Dado un juego de etapa G,

1. Si G tiene un único EN $\sigma = (\sigma_1, \ldots, \sigma_n)$, entonces para cualquier T natural y cualquier factor de descuento δ, el juego repetido $G^T(\delta)$ tiene un único ENPS consistente en el perfil de estrategias en el que cada jugador i elige incondicionalmente σ_i en cada etapa.

2. Si todos los EN de G conducen a los mismos pagos, entonces para cualquier T natural y cualquier factor de descuento δ, cualquier ENPS de $G^T(\delta)$ determina que en cada etapa se juegue un EN de G.

Dem. Basta con demostrar (2), pues (1) es un caso particular. Sea $\sigma = (\sigma_1, \ldots, \sigma_n)$ un ENPS y razonemos por inducción hacia atrás.

Siguiendo la demostración del Teorema 1.36, en cualquier subjuego iniciado en la última etapa, σ determina un EN de G. Por tanto, el pago obtenido por cada jugador i en dicha etapa es independiente de la T-historia seguida y, en consecuencia, la optimalidad de σ_i frente a σ_{-i} es equivalente a la optimalidad de la restricción de σ a la última etapa del juego $G^{T-1}(\delta)$, que concluye en la penúltima etapa de $G^T(\delta)$. Considerando el mismo razonamiento para $G^{T-1}(\delta)$ se obtiene que en la penúltima etapa de $G^T(\delta)$, σ también determina un EN. La iteración del razonamiento anterior conduce a la conclusión de que σ ha de determinar un EN en cualquier etapa. $\square$

Nota. *No se ha desarrollado la teoría relativa a juegos con información imperfecta, pero este mismo teorema se puede demostrar para tales juegos. Esto se debe a la existencia de un algoritmo de inducción hacia atrás generalizado, aplicable a juegos carentes de información perfecta.*

Nótese que cuando consideremos el juego repetido infinitamente, el teorema anterior deja de ser válido, pues no se puede tomar una última etapa del juego y razonar regresivamente para todas las demás.

Corolario 1.44. El dilema del prisionero finitamente repetido $\mathrm{DP}^T(\delta)$ tiene un único ENPS consistente en el perfil de estrategias incondicionadas (*Confesar siempre*, *Confesar siempre*).

1.5. Dilema del prisionero infinitamente repetido

Antes de comenzar a estudiar el DP cabe destacar que un mismo juego de etapa finitamente repetido es conceptualmente muy distinto del caso infinito. En el juego repetido un número concreto de veces los jugadores conocen cuál es el final del mismo y, por tanto, orientarán sus estrategias a conseguir el máximo beneficio acumulado tras la etapa final. En un juego iterado infinitamente los jugadores desconocen el final del mismo y, por tanto, han de actuar sin esa capacidad de previsión que afecta a las amenazas o promesas que puedan hacerse a través de sus distintas elecciones. Cabe destacar que un juego iterado, cuyo final depende de la aleatoriedad, es un juego repetido infinitamente. Es decir, si se juega al dilema del prisionero repetido y, al final de cada etapa, se lanza un dado al aire que determinará si el juego concluye o se continúa con la siguiente iteración, propiamente es un juego repetido infinitamente pues los jugadores desconocen cuál será la etapa final.

Ahora sí, volviendo al DP, se ha comprobado que la cooperación es el resultado deseable desde el punto de vista del conjunto de jugadores en el dilema del prisionero pero no es un EN. De hecho, el único EN es el comportamiento egoísta de ambos jugadores, que es además un EN en estrategias estrictamente dominantes. Por tanto, no parece previsible que se produzca cooperación.

Se ha comprobado en el Teorema 1.43 que, tomando como juego de etapa un juego con las características descritas en el párrafo anterior, el juego repetido finitamente no puede alcanzar un resultado en equilibrio en el que se dé la cooperación etapa por etapa. Por el contrario, en contraposición al Corolario 1.44, en esta sección se comprobará que en el dilema del prisionero repetido infinitamente existen múltiples ENPS en los que se da cooperación en cada etapa.

Definición 1.45. Sea un juego de etapa cualquiera $G = \{A_1, \ldots, A_n;\, g_1, \ldots, g_n\}$, donde los pagos están acotados ($\exists M \in \mathbb{R}$ tal que $|g_i(a_1, \ldots, a_n)| < M$ para todo i y todo $(a_1, \ldots, a_n) \in A$). Sea $\delta = (\delta_1, \ldots, \delta_n)$ un vector de factores de descuento, donde $\delta_i < 1, \forall i$. Llamamos juego repetido infinitamente $G^\infty(\delta)$ al juego que cumple, siendo de dominio público, que

1. Antes de comenzar a jugar
   i) El factor de descuento de cada jugador i es δ_i
   ii) Tras cualquier etapa k el juego puede proseguir en la etapa siguiente.
   iii) Los pagos de $G^\infty(\delta)$ son, para cada jugador i, el valor presente, para el factor de descuento δ_i, de la sucesión infinita de sus pagos de etapa.
2. Antes de que empiece cualquier nueva etapa, son de dominio público las jugadas realizadas en todas las etapas anteriores.

En lo sucesivo se supondrá que $\delta_i = \delta$ y, abusando de la notación, $G^\infty(\delta)$ será el juego repetido de cada jugador. Se considerará así mismo que G es un juego en forma estratégica.

A continuación se adaptarán ciertos conceptos ya utilizados a los juegos infinitamente repetidos. Sea el juego de etapa $G = \{A_1, \ldots, A_n;\, g_1, \ldots, g_n\}$.

Definición 1.46. Las t-historias del juego repetido $G^\infty(\delta)$ se definen de manera idéntica al caso finito.

Definición 1.47. Las estrategias σ_i del jugador i son los planes que determinan qué acción realizará dicho jugador en cada etapa para cada posible historia dada hasta ese momento. Se describen como $\sigma_i = (a_i^1(h), a_i^2(h), \ldots, a_i^t(h), \ldots)$, donde $a_i^t(h)$ son funciones definidas como en el caso finito.

Igual que ocurre en el caso finito, un perfil de estrategias (puras o mixtas) $\sigma = (\sigma_1, \ldots, \sigma_n)$ del juego repetido $G^\infty(\delta)$ determina la sucesión de combinaciones de acciones que pondrán en práctica los jugadores para cada etapa t dada por

$$\left\{\left(a_1^t(h_t(\sigma)), a_2^t(h_t(\sigma)), \ldots, a_n^t(h_t(\sigma))\right)\right\}_{t=1,2,\ldots}, \tag{1.32}$$

donde, precisamente, $h_t(\sigma) = \{(a_1^m(h_m(\sigma)), a_2^m(h_m(\sigma)), \ldots, a_n^m(h_m(\sigma)))\}_{m=1,\ldots,t-1}$. Esta sucesión constituye la trayectoria del juego determinada por σ. Así pues, los pagos globales se determinan a partir de los pagos de etapa de acuerdo con el factor de descuento.

Definición 1.48. Los subjuegos de $G^\infty(\delta)$ son cada uno de los juegos que inician en la etapa t dada una de las t-historias posibles y tienen idéntica estructura al juego global.

A partir de ahora se utilizará la notación $\mathrm{DP}^\infty(\delta)$ para referirse al dilema del prisionero repetido infinitamente. Cabe además destacar que, al consultar la literatura, en la mayoría de ocasiones se hace referencia a $\mathrm{DP}^\infty(\delta)$ como dilema repetido o iterado "a secas", en lugar de denominar así al $\mathrm{DP}^T(\delta)$.

Para comenzar con el análisis veremos dos perfiles que no son EN.

Proposición 1.49. El perfil de cooperación (*Callar siempre*,*Callar siempre*) no es EN en $\mathrm{DP}^\infty(\delta)$.

Dem. El perfil de cooperación (*Callar siempre*,*Callar siempre*) determina una trayectoria de cooperación completa del juego consistente en (*Callar*,*Callar*) en cada etapa. Pero la mejor respuesta a "*Callar siempre*" de J1 es claramente "*Confesar siempre*" por parte de J2 independientemente del valor de δ, luego el perfil estudiado no es EN. □

Proposición 1.50. El perfil $\sigma = (\sigma_1, \sigma_2)$ en el que $\sigma_1 = \sigma_2 =$"*en la primera etapa jugar Confesar y para $t > 1$ jugar la acción del contrario en la etapa anterior*" no es EN en $\mathrm{DP}^\infty(\delta)$ para valores suficientemente altos de δ.

Dem. Dado σ_1 de J1, la estrategia σ_2 de J2 reporta a este último un flujo de pagos (1, 1, ...) con un valor presente $1/(1-\delta)$. En cambio, la estrategia $\sigma_2' =$"*Callar siempre*" reporta un valor presente $3\delta/(1-\delta)$, que es estrictamente mayor que $1/(1-\delta)$ siempre que $\delta > 1/3$. □

1.5.1. Perfiles de estrategias que son EN

Cónsiderese la estrategia ojo por ojo (*OPO*), también conocida como *TFT* por sus siglas en inglés. La estrategia *OPO* se define como: "*en la primera etapa jugar Callar y para $t > 1$ jugar la acción que el contrario jugó en la etapa anterior*".

El perfil $\sigma = (\sigma_1, \sigma_2)$ en el que $\sigma_1 = \sigma_2 = OPO$, determina una trayectoria de cooperación completa del juego consistente en (*Callar*,*Callar*) en cada etapa. Por otra parte;

Proposición 1.51. El perfil σ es un EN siempre que el valor de δ sea suficientemente alto, pero no es un ENPS.

Dem. Comprobémoslo por partes:

- σ es EN. Veamos que, en efecto, $\sigma_1 = OPO$ es una respuesta óptima por parte de J1 a $\sigma_2 = OPO$ de J2 (y viceversa) pues seguir σ_1 garantiza un pago de 3 en cada etapa mientras que cualquier desviación de σ_1 perjudicaría a J1. Comprobemos que esto es cierto.

  Si J1 se desvía de la acción *Callar* en alguna etapa consigue un pago de 5 a costa de ser castigado y no ser perdonado hasta que juegue *Callar*. Supongamos que J1 rectifica inmediatamente después, entonces el turno en el que ha jugado *Confesar* recibirá un pago de 5 mientras que al turno siguiente recibirá 0. Comparando los valores presentes;

  $$\begin{array}{ll} \text{- Con desviación:} & v' = 5 + \delta \cdot 0 = 5, \\ \text{- Sin desvíación:} & v = 3 + 3 \cdot \delta, \end{array}$$

  la estrategia sin desviación es mejor siempre que $3 + 3 \cdot \delta \geq 5$, es decir, si $\delta \geq 2/3$.

  Supongamos ahora que la desviación consta de $k + 2$ etapas siendo $k \geq 1$. J1 obtendría los pagos:

Turnos	1	2	...	k+1	k+2
Pago con desviación	5	1	...	1	0
Pago sin desviación	3	3	...	3	3

  Y podemos comparar los valores presentes

  $$\begin{array}{ll} \text{Con desviación} & v' = 5 + 1\delta + \cdots + 1\delta^n + 0 = 5 + \delta(\delta^n - 1)/(\delta - 1) \\ \text{Sin desviación} & v = 3 + 3\delta + \cdots + 3\delta^n + 3\delta^{n+1} = 3(\delta^{n+2} - 1)/(\delta - 1), \end{array}$$

que indican que σ_1 es mejor siempre que

$$\begin{aligned} 3(\delta^{n+2}-1)/(\delta-1) &\geq 5+\delta(\delta^n-1)/(\delta-1) \\ 3\delta^{n+2}-3 &\leq 4\delta-5+\delta^{n+1} \\ 3\delta^{n+2}-\delta^{n+1}-4\delta+2 &\leq 0. \end{aligned} \tag{1.33}$$

Por último supongamos que J1 se desvía y juega *Confesar* indefinidamente. Entonces los valores presentes son:

Con desviación $v' = 5 + 1\delta + \cdots + 1\delta^n + \cdots = 5 + \delta/(1-\delta)$
Sin desviación $v = 3 + 3\delta + \cdots + 3\delta^n + \cdots = 3/(1-\delta)$

y σ_1 es mejor siempre que $3/(1-\delta) \geq 5 + \delta/(1-\delta)$, es decir $\delta \geq 1/2$. En conclusión, el perfil de estrategias *OPO* es un EN si $\delta \geq 2/3$.

- σ no es ENPS. Tómese el subjuego cuyo resultado de etapa anterior es (*Callar*, *Confesar*). A J1 le conviene desviarse de σ_1 si J2 va a seguir σ_2, pues si no se desviara produciría una cadena infinita dada por (*Confesar*, *Callar*),(*Callar*, *Confesar*),(*Confesar*, *Callar*),(*Callar*, *Confesar*),... que determina los pagos

$$5 + 0\cdot\delta + 5\cdot\delta^2 + 0\cdot\delta^3 + 5\cdot\delta^4 + \cdots = 5(1+\delta^2+\delta^4+\dots) = \frac{5}{1-\delta^2}. \tag{1.34}$$

Mientas que si eligiera *Callar* en adelante se obtendría la cadena (*Callar*, *Callar*),(*Callar*, *Callar*),... con pagos

$$3 + 3\cdot\delta + 3\cdot\delta^2 + 3\cdot\delta^3 + \cdots = 3(1+\delta+\delta^2+\delta^3+\dots) = \frac{3}{1-\delta}. \tag{1.35}$$

Así pues, el perfil *Callar* reporta un pago mayor siempre y cuando

$$\frac{3}{1-\delta} \geq \frac{5}{1-\delta^2} \Longrightarrow \delta \geq 2/3. \tag{1.36}$$

Con lo cual, para valores $\delta < 2/3$ el perfil de estrategias *OPO* no es EN y, para valores $\delta \geq 2/3$, es EN pero no ENPS. □

Proposición 1.52. El perfil de estrategias incrédulas ($\sigma = (\sigma_1, \sigma_2)$ con $\sigma_1 = \sigma_2 =$ "*Confesar siempre*") es ENPS.

Dem. σ es EN, pues es evidente que σ_1 es la respuesta óptima a σ_2 (y viceversa). Basta con notar que la restricción de σ a cualquier subjuego resulta en el mismo perfil, luego es EN en cualquier subjuego y por tanto ENPS. □

Conviene añadir a los dos anteriores el perfil de estrategias de disparador dado por la estrategia

ED = "*Callar en la primera etapa y callar en la etapa t si la historia del juego hasta ese momento ha sido siempre Callar por parte de ambos jugadores; Confesar en caso contrario*".

El perfil $\sigma = $ (ED,ED) es EN y ENPS si el factor de descuento es suficientemente alto. Si bien no se dará una demostración específica de lo anterior en este texto, se puede encontrar una en [3].

Los perfiles de estrategias que hemos visto indican que la repetición infinita del dilema del prisionero favorece la aparición de nuevos EN y ENPS donde algunos sustentan la cooperación de manera indefinida. Esto parece paradójico teniendo en cuenta que el único EN del juego de etapa es un equilibrio en estrategias dominantes determinado por la no cooperación de ambos jugadores.

Por otro lado, se han encontrado dos ENPS y, de hecho, hay más. Esto lleva a pensar que es posible que exista una cantidad de ENPS muy grande, entre los que sea difícil elegir uno. El Teorema 1.53 permite averiguar qué vectores de pagos son alcanzables por medio de ENPS y, con ello, obtener las posibilidades de mejora en el sentido de Pareto. Dicho teorema responde a la idea intuitiva de que, si los jugadores son suficientemente pacientes, es decir, no valoran demasiado la inmediatez del pago, son óptimas las estrategias de cooperación en las que se contempla un castigo indefinido a quien traicione la confianza del resto.

Teorema 1.53 (Teorema de tradición oral de Friedman). Sea G un juego finito, estático y con información completa. Sea el perfil $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*)$ un EN de G y sea $g(\alpha^*) = (g_1(\alpha^*), g_2(\alpha^*), \ldots, g_n(\alpha^*))$ el vector de pagos correspondiente a α^*. Si $v = (v_1, v_2, \ldots, v_n)$ es un vector de pagos factible de G que cumple $v_i > g_i(\alpha^*)$ para cualquier jugador i, entonces existe un factor de descuento δ_0 tal que para todo $\delta > \delta_0$ existe un ENPS de $G^\infty(\delta)$ que alcanza v como vector de pagos medios.

Dem. Considérese primero que existe un perfil $\alpha = (\alpha_1, \ldots, \alpha_n)$ del juego de etapa G que produce el vector de pagos v. Recordemos que esto no ocurre necesariamente para un vector de pagos factible cualquiera.

Sea $\sigma = (\sigma_1, \ldots, \sigma_n)$ el perfil de $G^\infty(\delta)$ en el que, para cada jugador i,

$\sigma_i =$ "*Jugar α_i en la primera etapa. En cualquier otra etapa jugar α_i siempre que se haya jugado el perfil de etapa α en todas las etapas anteriores; jugar α_i^* en caso contrario*".

Veamos que σ es EN de $G^\infty(\delta)$ si δ es suficientemente grande.

Supongamos fijado σ_{-i}. Entonces, si en alguna etapa algún jugador se ha desviado del perfil α (i ha sido el primero en hacerlo necesariamente), en adelante, $\sigma_i = \alpha_i^*$ es respuesta óptima a $\sigma_{-i} = \alpha^*_{-i}$ por parte de i. Supongamos ahora que nos encontramos en la primera etapa o en una en la que no ha habido desviación previa del perfil α. Veamos en este caso que, para ciertos δ, σ_i es respuesta óptima a σ_{-i}.

Sea α'_i la respuesta óptima a α_{-i} para un jugador i en una etapa t en la que no se han producido aún desviaciones y sea d_i el beneficio obtenido por i al adoptar dicha desviación. Como consecuencia, el resto de jugadores adoptarán α^*_{-i} en adelante como castigo y el jugador i se verá obligado a jugar su estrategia óptima α^*_i en consecuencia. Así pues, el flujo de pagos óptimo que obtendría i en caso de desviarse viene dado por la sucesión

$$\{r_m\}_{m=0}^{\infty} = \begin{cases} g_i(\alpha'_i, \alpha_{-i}) = v_i + d_i & \text{para } m = 0 \\ g_i(\alpha^*) & \text{para } m = 1, 2, \ldots \end{cases}, \tag{1.37}$$

cuyo valor presente correspondiente es

$$\mathrm{VP}' = \sum_{m=0}^{\infty} \delta^m r_m = v_i + d_i + \sum_{m=1}^{\infty} \delta^m g_i(\alpha^*) = v_i + d_i + \delta g_i(\alpha^*)/(1-\delta), \tag{1.38}$$

pues recordemos que $\delta < 1$. Pero considerando el caso en el que i no se desviara, el flujo de pagos sería la sucesión constante $\{g(\alpha)\}_{m=1,2,\ldots} = \{v_i\}_{m=1,2,\ldots}$ y el valor presente correspondiente

$$\mathrm{VP} = \sum_{m=0}^{\infty} \delta^m v_i = v_i/(1-\delta). \tag{1.39}$$

Recapitulando, σ_i es respuesta óptima de i a σ_{-i} si $\mathrm{VP} \geq \mathrm{VP}'$, es decir, si

$$\begin{aligned} v_i/(1-\delta) &\geq v_i + d_i + \delta g_i(\alpha^*)/(1-\delta) \\ 0 &\geq -\delta v_i + d_i - \delta d_i + \delta g_i(\alpha^*) \\ \delta &\geq \frac{d_i}{v_i + d_i - g_i(\alpha^*)}, \end{aligned} \tag{1.40}$$

lo cual implica que σ es EN de $G^{\infty}(\delta)$ siempre que $\delta \geq \max_{i \in J}\{d_i/(v_i + d_i - g_i(\alpha^*))\}$.

Ahora bien, todos los subjuegos de $G^{\infty}(\delta)$ tienen la misma estructura que el propio $G^{\infty}(\delta)$ y parten, o bien de historias en las que se ha producido alguna desviación con respecto al perfil α, o bien de aquellas en las que ningún jugador se ha desviado. Ambos casos los acabamos de estudiar, con lo cual en todos los subjuegos la estrategia σ es EN y, por tanto, se concluye que el teorema se cumple tomando

$$\delta_0 = \max_{i \in J}\{d_i/(v_i + d_i - g_i(\alpha^*))\}. \tag{1.41}$$

Queda considerar el caso en el que el vector v no corresponda al pago esperado de ninguna estrategia pura o mixta. Dicha parte de la demostración se puede consultar en la fuente original (ver [4]). En la citada publicación se da una demostración del teorema completo utilizando un volumen de resultados previos cuya inclusión en este trabajo no se ha estimado necesaria en relación a los objetivos del mismo. $\square$

Nota. *En la práctica es más rápido maximizar la expresión* $(\rho_i - v_i)/(\rho_i - g_i(\alpha^*))$ *en la ecuación (1.41), donde* ρ_i *es el mejor pago al que puede optar* i *al desviarse de* α.

La interpretación del Teorema 1.53 es que si un vector de pagos factible v da, para todos los jugadores, pagos estrictamente mayores que el vector de pagos asociado a un perfil que constituye un EN, entonces existe un ENPS del juego repetido infinitamente cuyo vector de pagos es v para todo δ suficientemente próximo a 1. Esto se consigue tomando por todos los jugadores como estrategia inicial aquella que da el vector de pagos v y, en caso de que un jugador se desvíe, todos los demás adoptarán como estrategia aquella que minimice los pagos de dicho jugador para siempre. Esto justifica que el perfil de estrategias de disparador sea ENPS en $\mathrm{DP}^{\infty}(\delta)$.

2. Autómatas

2.1. Fundamentos de las máquinas secuenciales estocásticas

2.1.1. Introducción. Definiciones y autómatas deterministas

En este epígrafe se presentarán, en primer lugar, dos ejemplos que permitirán introducir el concepto de autómata y, posteriormente, construir los autómatas estocásticos.

Definición 2.1. Un autómata o máquina de Moore es una 6-upla $(S, s_0, X, Y, \delta, G)$ cuyos elementos son:

- Un conjunto finito S de estados.
- Un estado inicial $s_0 \in S$
- Un conjunto finito X denominado *alfabeto input*.
- Un conjunto finito Y denominado *alfabeto output*.
- Una función de transición $\delta : S \times X \longrightarrow S$ que asigna a cada par (s, x) el siguiente estado.
- Una función de output $G : S \longrightarrow Y$ que envía un estado en un output.

Nota. *Para el resto del TFG se utilizará input y output para hacer referencia a un elemento del alfabeto input y del alfabeto output respectivamente.*

Si A es un conjunto, denominado alfabeto, una palabra sobre el alfabeto A es una sucesión $a_1, a_2, \ldots, a_n$ donde cada entrada a_k es un elemento de A. Los elementos de A se denominan símbolos.

Estos autómatas descritos se desarrollaron originalmente en [13]. La Definición 2.1 responde al concepto de una máquina de $n = |S|$ estados con sus correspondientes respuestas (outputs) asociadas, inicializada en uno de ellos. Por otro lado, la transición de un estado a otro viene dada por el estímulo (input) que recibe el autómata en cada instante.

La Figura 2.1 es la representación gráfica de un autómata de Moore. Los estados están representados por cada uno de los círculos y su output es el contenido del mismo. Las flechas representan las transiciones entre estados, etiquetadas con el input que conduce a cada una de ellas.

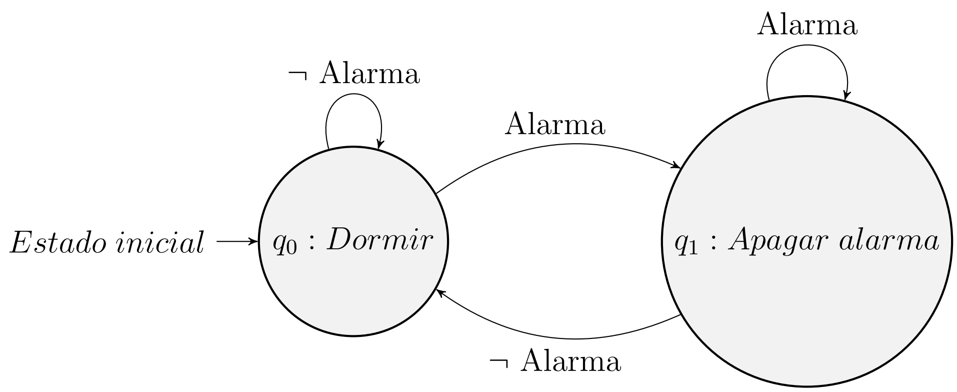

Figura 2.1: Autómata de Moore que representa un procrastinador infinito.

Definición 2.2. Un autómata o máquina de Mealy es una 6-upla $(S, s_0, X, Y, \delta, G)$ cuyos elementos son:

- Un conjunto finito S de estados.
- Un estado inicial $s_0 \in S$
- Un conjunto finito X denominado *alfabeto input*.
- Un conjunto finito Y denominado *alfabeto output*.
- Una función de transición $\delta : S \times X \longrightarrow S$ que envía un par estado-input en el siguiente estado.
- Una función de output $G : S \times X \longrightarrow Y$ que envía un par estado-input en un output.

Estos autómatas se desarrollaron originalmente en [11]. La Definición 2.2 describe el concepto de una máquina de $n = |S|$ estados, inicializada en uno de ellos en la cual, tanto la transición entre estados como el output de cada uno viene dado por el input. La Figura 2.2 es la representación gráfica de un autómata de Mealy equivalente al de la Figura 2.1. En este caso las transiciones son las que tienen asociada una acción en particular y no los estados, así pues, sonar la alarma conlleva que el autómata la apague y vuelva al estado q_0, donde esperará a recibir un nuevo input.

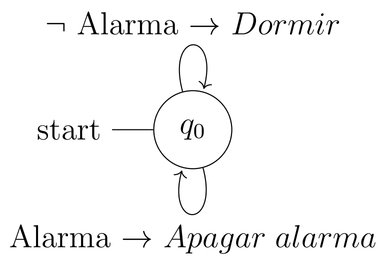

Figura 2.2: Autómata de Mealy que representa un procrastinador infinito.

Así pues, la diferencia fundamental entre ambos conceptos es que en el autómata de Moore el output depende únicamente del estado actual mientras que en el de Mealy el output también depende del input recibido.

2.1.2. Autómatas secuenciales estocásticos

El grueso de este epígrafe se basa en el primer capítulo de [14] combinado con demostraciones propias de varios resultados.

A continuación se generalizan los conceptos anteriormente expuestos considerando que las transiciones entre estados no sean deterministas, sino que vengan dadas por un vector de probabilidades.

Definición 2.3. Una máquina secuencial estocástica de tipo Mealy (SSM de Mealy) es una 4-upla $M = (S, X, Y, \{A(y|x)\})$ cuyos elementos son:

- Un conjunto finito S de estados.
- Un conjunto finito X denominado *alfabeto input*.
- Un conjunto finito Y denominado *alfabeto output*.
- Un conjunto $\{A(y|x)\}$ cuyo contenido son $|X|\cdot|Y|$ matrices cuadradas de orden $|S|$ tales que

$$a_{ij}(y|x) \geq 0 \ , \ \forall i, j$$

$$\sum_{y\in Y}\sum_{j=1}^{|S|} a_{ij}(y|x) = 1,$$

donde $A(y|x) = [a_{ij}(y|x)]$.

Supuesta una distribución probabilística inicial sobre los estados de la máquina dada por un vector π, y dada una secuencia de inputs representada por la palabra $u = x_1 x_2 \cdots x_k$, entonces la SSM de Mealy devuelve el output $v = y_1 y_2 \cdots y_k$ y avanza al siguiente estado. La transición está dada por las matrices $A(y|x)$, donde $a_{ij}(y|x)$ es la probabilidad de la máquina de pasar del estado s_i al estado s_j y devolver el output y cuando se encuentra en el estado s_i y recibe el input x.

La siguiente proposición explica la relación que guardan los elementos anteriores.

Proposición 2.4. Sea $u = x_1 x_2 \cdots x_k$ una palabra en el alfabeto X y $v = y_1 y_2 \cdots y_k$ del alfabeto Y, considérese

$$A(v|u) = [a_{ij}(v|u)] = A(y_1|x_1)A(y_2|x_2)\cdots A(y_k|x_k). \tag{2.1}$$

Entonces $a_{ij}(v|u)$ es la probabilidad de que la máquina pase del estado s_i al estado s_j y devuelva la secuencia de outputs v dada la secuencia de inputs u.

Dem. La proposición es cierta en el caso $l(v,u) = 1$, donde $l(v,u)$ denota la longitud común de ambas palabras. Esto es trivial, pues $(v,u) = (y,x)$. Ahora bien, asumiendo cierta la proposición para $L(v,u) = k-1$, se tiene que

$$a_{ij}(vy|ux) = \sum_k a_{ik}(v|u)a_{kj}(y|x). \tag{2.2}$$

Puesto que la parte derecha de la igualdad es precisamente la probabilidad de que la máquina pase del estado s_i al s_j devolviendo el output vy dado el input ux, la proposición queda probada por inducción para todos u,v con $l(v,u) \geq 1$.

Por otro lado, siendo λ la palabra vacía ($l(v,u) = l(\lambda,\lambda) = 0$) se define $A(\lambda|\lambda) = I$. Dado que I es la matriz unitaria $|S|$-dimensional, no se producirá cambio alguno en el estado del autómata ni se devolverá output cuando no se introduzca input alguno. □

Antes de continuar con las máquinas de Moore se darán algunas definiciones más. Denotaremos por η al vector columna cuyos elementos son todos iguales a 1 y tiene dimensión igual al número de estados de la máquina a la que hace referencia.

Definición 2.5. Dado un autómata M y un par input-output (v,u), se define

$$\eta(v|u) = A(v|u)\eta \quad (\eta(\lambda,\lambda) = I\eta = \eta)\,. \tag{2.3}$$

Nótese que el i-ésimo elemento del vector $\eta(v|u)$ es la suma de los elementos de la i-ésima fila de la matriz $A(v|u)$. Por tanto $\eta(v|u)$ recoge las probabilidades de que, dada una secuencia de inputs u, se partiera del estado s_i si se devuelve la secuencia v.

Dado que estamos considerando una máquina cuyas transiciones entre estados son estocásticas, cabe también considerar que la determinación del estado inicial tenga la misma naturaleza. La siguiente definición formaliza lo anterior:

Definición 2.6. Sea π un vector probabilístico que determina la distribución de probabilidad inicial sobre los estados del autómata. Dado un par input-output (v, u) de palabras, se definen el vector $\pi(v|u)$ y la función $p_\pi(v|u)$ como sigue:

$$\pi(v, u) = \pi A(v|u) \quad (\pi(\lambda, \lambda) = \pi I = \pi) \tag{2.4}$$

$$p_\pi(v|u) = \pi\eta(v|u) \quad (= \pi(v, u)\eta)\,. \tag{2.5}$$

Así definida, $p_\pi(v|u)$ es la probabilidad de que la máquina devuelva la palabra v cuando su distribución inicial es π y se le introduce la palabra u. Por otro lado, la j-ésima componente del vector $\pi(v, u)$ se corresponde con la probabilidad de que el autómata devuelva v y se encuentre en el estado s_j dados el input u y la distribución inicial π. Nótese también que

$$\begin{aligned} p_\pi(v_1v_2, u_1u_2) &= \pi\eta(v_1v_2, u_1u_2) = \pi A(v_1v_2, u_1u_2)\eta = \\ &= \pi A(v_1, u_1)A(v_2, u_2)\eta = \pi(v_1, u_1)\eta(v_2, u_2). \end{aligned} \tag{2.6}$$

Definición 2.7. Para cada distribución inicial π se considera el vector $\bar{\pi}(v, u)$ de componentes $\bar{\pi}_j(v, u)$ tales que

$$\bar{\pi}_j(v, u) \cdot p_\pi(v|u) = \pi_j(v|u). \tag{2.7}$$

Es importante reseñar la diferencia entre $\bar{\pi}(v, u)$ y $\pi(v, u)$. El segundo vector no es probabilístico y sus componentes son las probabilidades de que el autómata entre en el estado j-ésimo y se devuelva el output v cuando el input es u y la distribución de probabilidad inicial es π. Pero dado dicho input y dicha distribución, pueden existir otros outputs con probabilidad estrictamente positiva. Por el contrario, en el vector $\bar{\pi}(v, u)$ se ha asumido de antemano que el output obtenido es v.

Por último, nótese que el modelo dado generaliza el concepto de autómata determinista de Mealy. Una SSM de Mealy en la que todas las filas de cada matriz $\{A(y|x)\}$ son vectores de la base canónica es precisamente una máquina de Mealy determinista. Así pues, dado un autómata de Mealy $M(S, s_0, X, Y, \delta, G)$ basta considerar los mismos estados y alfabetos de input y output para definir una SSM equivalente: cada matriz $A(y|x)$ es una matriz nula excepto por los elementos $a_{ij}(y|x)$ tales que $\delta(s_i, x) = s_j$ y $G(s_i, x) = y$, que valen 1.

Definición 2.8. Una máquina secuencial estocástica de tipo Moore (SSM de Moore) es una 5-upla $M = (S, X, Y, \{A(x)\}, \Lambda)$ cuyos elementos son:

- Un conjunto finito S de estados.
- Un conjunto finito X denominado *alfabeto input*.
- Un conjunto finito Y denominado *alfabeto output*.
- Un conjunto $\{A(x)\}$ cuyo contenido son $|X|$ matrices estocásticas cuadradas de orden $|S|$.
- Una función $\Lambda : S \longrightarrow Y$ determinista.

Nota. *Se considera una matriz estocástica aquella cuyos coeficientes son no negativos y la suma de los elementos de cada fila es la unidad.*

Análogamente al caso de Mealy, dada una distribución inicial π, los valores $a_{ij}(x)$ representan la probabilidad de que la máquina transicione del estado s_i al estado s_j dado el input x. Además, cuando la máquina se encuentra en el estado s_j, la máquina devuelve el output $\Lambda(s_j) \in Y$.

Proposición 2.9. Sea u una palabra en el alfabeto X y $A(u)$ definida como

$$A(u) = \begin{cases} I & si\ u = \lambda \\ [a_{ij}(u)] = A(x_1)A(x_2)\cdots A(x_k) & si\ u = x_1x_2\cdots x_k \end{cases}. \tag{2.8}$$

Entonces $a_{ij}(u)$ es la probabilidad de que la máquina pase del estado s_i al estado s_j dada la secuencia de inputs u.

Dem. Análogamente a lo expuesto a continuación de la Definición 2.1 podemos ver que cada elemento

$$a_{ij}(ux) = \sum_k a_{ik}(u)a_{kj}(x), \tag{2.9}$$

que, según la interpretación que se ha dado de la definición de SSM de Moore, se corresponde con la probabilidad de que la máquina avance al estado s_j partiendo del estado s_i dada la secuencia de inputs ux. Nótese que los elementos $a_{ik}(u)$ son las probabilidades de que el autómata haya transicionado al estado s_k partiendo del s_i cuando se ha introducido la secuencia de inputs u. Por otro lado, $a_{ki}(x)$ son las probabilidades de que dado el estado s_k la máquina transicione al s_i cuando se introduce el input x.

Basta reconocer que $A(\lambda)$ indica que, si no se introduce input alguno, el autómata permanece en su estado actual. $\square$

Definición 2.10. Dos autómatas M y M' son equivalentes estado a estado si para cada estado s_i de M existe un estado correspondiente s_j de M' tal que $p_{s_i}^M(v|u) = p_{s_j}^{M'}(v|u)$ para todo par input-output (v, u) con $l(v, u) \geq 1$ y también se cumple en sentido inverso, entre M' y M.

Antes de introducir el teorema que garantiza la equivalencia de las dos descripciones de SSM dadas conviene hacer una reseña sobre notación. De la misma forma que $p_\pi(v|u)$ es la probabilidad de que la máquina devuelva la palabra v cuando su distribución inicial es π y se le introduce la palabra u, cometiendo un abuso de lenguaje $p_{s_i}(v|u)$ lo es cuando la máquina parte del estado inicial s_i. Por otro lado, $p_{ij}(x)$ es la probabilidad de que el autómata pase del estado s_i al estado s_j cuando recibe x como input en una SSM de Moore y $p_{ij}(y|x)$ es la probabilidad de que lo haga en una SSM de Mealy devolviendo el output y.

Teorema 2.11. Toda SSM de Moore tiene una SSM de Mealy equivalente estado a estado y viceversa.

Dem. Probemos por construcción la existencia de dichos autómatas.

$\rightarrow$ Sea $M = (S, X, Y, \{A(x)\}, \Lambda)$ una SSM de Moore. Veamos que la SSM de Mealy $M' = (S, X, Y, \{A'(y|x)\})$ es equivalente a M estado a estado, estando $A'(y|x) = [a'_{ij}(y|x)]$ definida mediante

$$a'_{ij}(y|x) = \begin{cases} a_{ij}(x), & si \;\; y = \Lambda(s_j) \\ 0, & \text{en caso contrario} \end{cases} . \tag{2.10}$$

Se procederá por inducción en l. Sea $\mathrm{In}(\Lambda^{-1}(y))$ el conjunto de índices de los estados s_j tales que $\Lambda(s_j) = y$. Entonces

$$p_{s_i}^M(y|x) = \sum_{j \in \mathrm{In}(\Lambda^{-1}(y))} p_{ij}^M(x) = \sum_{j \in \mathrm{In}(\Lambda^{-1}(y))} a_{ij}(x) = \sum_{j=1}^{|S|} a'_{ij}(y|x) = p_{s_i}^{M'}(y|x), \tag{2.11}$$

donde la penúltima igualdad viene dada por el hecho de que los sumandos de ambos miembros son iguales salvo para los j no pertenecientes a $\mathrm{In}(\Lambda^{-1}(y))$, que dan lugar a elementos $a'_{ij}(y|x)$ nulos. Consideremos la hipótesis inductiva cierta para el par de secuencias input-output (u, v) y procedemos.

$$p_{s_i}^M(vy|ux) = \sum_{j=1}^{|S|} p_{s_i}^M(v|u) p_{s_j}^M(y|x) = \sum_{j=1}^{|S|} p_{s_i}^{M'}(v|u) p_{s_j}^{M'}(y|x) = p_{s_i}^{M'}(vy|ux). \tag{2.12}$$

La igualdad (2.11) implica que la correspondencia que envía estados de M en estados de M' ($s_i \leftrightarrow s'_i = s_i$) es la identidad. Finalmente, dado que la identidad es una aplicación biyectiva entre estados queda probado que para cada estado de M existe un estado correspondiente de M' y viceversa.

$\leftarrow$ Sea $M = (S, X, Y, \{A(y|x)\})$ una SSM de Mealy. Veamos que la SSM de Moore $M' = (S, X, Y, \{A'(x)\}, \Lambda)$ es equivalente a M estado a estado, siendo S' el producto cartesiano $S \times Y$, $\Lambda(s_i, y) = y$ y

$$A'(x) = \begin{bmatrix} A(y_1|x) & A(y_2|x) \dots A(y_k|x) \\ A(y_1|x) & A(y_2|x) \dots A(y_k|x) \\ \vdots & \vdots \\ A(y_1|x) & A(y_2|x) \dots A(y_k|x) \end{bmatrix}. \tag{2.13}$$

Consideremos el output indexado por el valor m y $J = \text{In}(\Lambda^{-1}(y_m))$. Partiendo de la máquina M y, con un razonamiento análogo al de la SSM de Moore,

$$p_{s_i}^M(y_m|x) = \sum_{j=1}^{|S|} p_{ij}^M(y_m|x) = \sum_{j=1}^{|S|} a_{ij}(y_m|x) =$$

$$= \sum_{j=1}^{|S|} a'_{i,j+|S|(m-1)}(x) = \sum_{j\in J} a'_{ij}(x) = \sum_{j\in J} p_{ij}^{M'}(x) = p_{s_i}^{M'}(y_m|x). \tag{2.14}$$

De nuevo, considerando ahora pares de secuencias input-output (u, v) procedemos por inducción igual que se ha hecho en el caso anterior:

$$p_{s_i}^M(vy|ux) = \sum_{j=1}^{|S|} p_{s_i}^M(v|u)p_{s_j}^M(y|x) = \sum_{j=1}^{|S|} p_{s_i}^{M'}(v|u)p_{s_j}^{M'}(y|x) = p_{s_i}^{M'}(vy|ux). \tag{2.15}$$

Es necesario hacer notar que el desarrollo anterior únicamente determina una aplicación inyectiva entre S y S'. Ahora bien, para todo estado s'_i de M' también existe su correspondiente estado de M, de hecho viene dado por s_j con $j = |S|$ cuando $j = 0 \bmod |S|$ y $j = i \bmod |S|$ en caso contrario. $\square$

El Teorema 2.11 garantiza que las dos descripciones de SSM dadas son equivalentes y, por tanto, se pueden usar indistintamente para hablar de propiedades generales de los autómatas. Nótese que ambas representaciones generalizan los principales tipos de autómata determinista: las máquinas de Moore y de Mealy. Cabe también señalar que existen otros tipos de máquinas secuenciales estocásticas que generalizan lo expuesto hasta ahora.

2.2. Teoría de estados

En esta sección se busca explorar el concepto de equivalencia entre autómatas. Esta exploración conducirá a intentar, dado un autómata, buscar autómatas "más pequeños" que tengan la misma función. También se introducirá el concepto de máquinas secuenciales estocásticas iniciadas, que será un objeto matemático fundamental posteriormente.

Antes de entrar en lo anteriormente citado es necesario introducir un par de conceptos que se utilizarán más adelante. Nótese que a partir de ahora, como consecuencia del Teorema 2.11, utilizaremos SSM para referirnos indistintamente a máquinas de Moore o Mealy.

Definición 2.12. Dada M una SSM, K^M denota el conjunto ordenado

$$K^M = (\eta^M(\lambda|\lambda), \dots, \eta^M(y|x), \dots, \eta^M(v|u), \dots), \tag{2.16}$$

tal que todos los vectores de la forma $\eta^M(v|u)$ están contenidos en él para todo par (v, u) y cuyo órden está inducido por algún orden lexicográfico fijo en los pares (v, u).

El conjunto anterior se puede representar también mediante una matriz infinita cuyas columnas son los vectores de K^M, que denotaremos $[K^M]$. Asimismo, utilizaremos $K^M(m)$ para referirnos al subconjunto de K^M cuyos vectores hacen referencia a pares tales que $l(u, v) \leq m$ y cuyo orden está inducido por el de K^M.

Proposición 2.13. Sean $\mathscr{S}(m)$ el espacio vectorial generado por los vectores de $K^M(m)$ y $\mathscr{S}$ el generado por los de K^M. Existe m_0 tal que $\dim \mathscr{S}(m_0) = \dim \mathscr{S}$.

Dem. Supóngase que existe un i para el cual $\mathscr{S}(i) \equiv \mathscr{S}(i+1)$. Consideremos el vector $\eta \in \mathscr{S}(i+2)$:

$$\begin{aligned} \eta &= \sum a_k \eta(v_k|u_k) && y \quad l(u_k, v_k) \leq i+2 \\ \Longrightarrow \eta &= \sum a_k A(y_k|x_k)\eta(v'_k|u'_k) && y \quad l(u'_k, v'_k) \leq i+1 \\ \overset{\mathscr{S}(i)\equiv\mathscr{S}(i+1)}{\Longrightarrow} \eta &= \sum_k a_k A(y_k|x_k) \sum_j b_j \eta(v''_{kj}|u''_{kj}) && y \quad l(u''_{kj}, v''_{kj}) \leq i, \end{aligned} \tag{2.17}$$

y podemos reescribir η como

$$\eta = \sum_k \sum_j a_k b_j A(y_k|x_k)\eta(v''_{kj}|u''_{kj}) = \sum_k \sum_j a_k b_j \eta(v'_{kj}|u'_{kj}) \quad \textit{con } l(u'_{kj}, v'_{kj}) \leq i+1. \tag{2.18}$$

Luego η es combinación lineal de vectores de $\mathscr{S}(i+1)$ y, por tanto, $\mathscr{S}(i+j) = \mathscr{S}(i)$ para todo j natural. Por otro lado, dados dos naturales tales que $i < j$ se tiene que o bien $\mathscr{S}(i) < \mathscr{S}(j)$ o bien $\mathscr{S}(i) \equiv \mathscr{S}(j)$ y es claro que $\dim \mathscr{S}(m) \leq n = |S|$ para todo m. Así pues, ha de existir un m_0 que cumpla

$$\mathscr{S}(0) \subsetneq \cdots \subsetneq \mathscr{S}(m_0) \equiv \cdots \equiv \mathscr{S}, \tag{2.19}$$

y por tanto $\dim \mathscr{S}(0) < \ldots < \dim \mathscr{S}(m_0) = \cdots = \dim \mathscr{S} \leq n$. □

Definición 2.14. Sea $\eta_1, \ldots, \eta_m$ la familia de vectores que cumple:

1. η_1 es el vector $\eta(\lambda|\lambda)$.

2. $\eta_1, \ldots, \eta_m$ son los primeros vectores de K^M que forman base de $\mathscr{S}$.

La matriz H^M se define como

$$H^M = [h_{ij}] = [\eta_1, \ldots, \eta_m], \quad i = 1, \ldots, n,\ j = 1, \ldots, m \leq n. \tag{2.20}$$

En lo sucesivo, cuando se haga referencia al rango de una SSM M nos referiremos al rango de la matriz H^M (que equivale a la dimensión de K^M).

2.2.1. Nociones sobre minimización de estados

Definición 2.15. Sean π y ρ dos distribuciones iniciales de una cierta SSM. Se dice que π y ρ son k-equivalentes si las funciones $p_\pi(v|u)$ y $p_\rho(v|u)$ toman los mismos valores para todos los pares (u, v) tales que $l(u, v) \leq k$.

Se dice que π y ρ son equivalentes si las funciones $p_\pi(v|u)$ y $p_\rho(v|u)$ toman los mismos valores para todos los pares (u, v).

Proposición 2.16. Dos distribuciones π y ρ de una cierta SSM son equivalentes si y solo si son $|S|$-equivalentes.

Dem. La implicación ($\Rightarrow$) es trivial. Tomemos entonces $n = |S|$ para aligerar notación y asumamos que $p_\pi(v|u) = p_\rho(v|u)$ para todos los (u, v) tales que $l(u, v) \leq n-1$. Entonces $\pi\eta(v|u) = \rho\eta(v|u)$ para todos los pares de longitud conjunta menor o igual que $n - 1$ y, por tanto, $\pi H^M = \rho H^M$ ya que todo vector $\eta(v|u)$ se puede escribir como combinación lineal de las m columnas n_i de H^M independientemente de la longitud de las palabras u y v. Concluimos que

$$p_\pi(v|u) = \pi\eta(v|u) = \pi \sum a_i\eta_i = \sum a_i\pi\eta_i = \sum a_i\rho\eta_i = \rho \sum a_i\eta_i = p_\rho(v|u). \tag{2.21}$$

$\square$

Corolario 2.17. Dos vectores iniciales π y ρ de una SSM M son equivalentes si y solo si $\pi H^M = \rho H^M$.

Lema 2.18. Sea M una SSM y sea $\xi_i(y|x)$ la i-ésima fila de la matriz $A(y|x)$, asumiendo que es una fila no nula. Sea ξ' un vector subestocástico ($\sum \xi_i \leq 1$, $x_i \geq 0$) tal que $\xi_i(y|x)H^M = \xi' H^M$. Entonces la SSM M' obtenida sustituyendo la fila $\xi_i(y|x)$ de $A(y|x)$ por $\xi' H^M$ es equivalente a M estado a estado.

Dem. Basta probar que $\eta^M(v|u) = \eta^{M'}(v|u)$, para cualquier par (u, v).

Si el par de símbolos (x, y) no aparece en (u, v) la igualdad anterior se satisface trivialmente. Consideremos por tanto que $(u, v) = (u_1 x u_2, v_1 y v_2)$ suponiendo que x no aparece en u_1 ni u_2 y que y tampoco lo hace en v_1 y v_2. Entonces

$$\eta^M(v|u) = \eta(v_1 y v_2|u_1 x u_2) = A^M(v_1|u_1)A^M(y|x)\eta^M(v_2|u_2). \tag{2.22}$$

Ahora bien, dado que $A^M(y|x)H^M = A^{M'}(y|x)H^M$ por definición y $\eta^M(v_2|u_2)$ es combinación lineal de las columnas de H^M se sigue que:

$$\begin{aligned}\eta^M(v|u) = A^M(v_1|u_1)A^M(y|x)\eta^M(v_2|u_2) &= A^M(v_1|u_1)A^{M'}(y|x)\eta^M(v_2|u_2) \\ &= A^{M'}(v_1|u_1)A^{M'}(y|x)\eta^{M'}(v_2|u_2) = \eta^{M'}(v|u).\end{aligned} \tag{2.23}$$

El paso de la primera a la segunda línea de la ecuación 2.23 está garantizado porque las palabras u_i y v_i no contienen a x e y respectivamente. El caso general se demuestra fácilmente por inducción en la cantidad de veces que aparece (x, y) en (u, v). □

Teorema 2.19. Sea M una SSM de n estados tal que dos filas de H^M son idénticas. Entonces se puede construir una SSM de $n - 1$ estados M^* equivalente a M estado a estado.

Dem. Sea ξ una fila de una matriz $A(y|x)$ de M y asumamos que las filas j y k de H^M son idénticas. Los coeficientes de ξ_j y ξ_k en el sumatorio $\sum \xi_i h_{iq}$ son iguales para $q = 1, 2, \ldots, m$. Así pues podemos definir M' cambiando el vector ξ de $A(y|x)$ por el vector ξ' tal que $\xi'_k = 0$, $\xi'_j = \xi_j + \xi_k$ y $\xi'_i = \xi_i$ para los i distintos de k y j. Entonces

$$\sum \xi_i h_{iq} = \xi_1 h_{1q} + \ldots + (\xi_j + \xi_k)h_{jq} + \ldots + 0h_{kq} + \ldots + \xi_m h_{mq} = \sum \xi'_i h_{iq}, \tag{2.24}$$

y por tanto M y M' son equivalentes estado a estado por el Lema 2.18. De esta forma podemos obtener la matriz M'' aplicando el proceso anterior a todas las columnas de todas las matrices de M y las nuevas matrices tendrán la columna k-ésima idénticamente nula. Basta con construir M^* eliminando las columnas y filas k-ésimas de todas las matrices de M''.

Las matrices de M^* están bien definidas, pues siguen cumpliendo que la suma de todos sus elementos es la unidad y dicha SSM es equivalente a M'' estado a estado considerando la correspondencia que envía a cada estado n-ésimo de M'' en el n-ésimo estado de M^* salvo s_k que se envía a s^*_j. La equivalencia estado a estado es transitiva, luego queda demostrado el teorema. □

Definición 2.20. Se dice que una SSM M es reducida si no existen dos filas iguales en H^M.

Corolario 2.21. Toda SSM tiene una SSM reducida equivalente estado a estado.

Dem. Es consecuencia directa de la Definición 2.20 y el Teorema 2.19. □

Definición 2.22. Una máquina secuencial estocástica iniciada (ISSM) es una dupla SSM con una distribución inicial fijada.

La noción de equivalencia de estos autómatas iniciados viene inducida por la equivalencia entre distribuciones iniciales y la equivalencia estado a estado.

Definición 2.23. Dos ISSM (M, π), (M^*, π^*) son k-equivalentes si $p_{\pi}^{M}(v|u) = p_{\pi^*}^{M^*}(v|u)$ para todos los (u, v) tales que $l(u, v) \leq k$.

Dos ISSM (M, π), (M^*, π^*) son equivalentes si $p_{\pi}^{M}(v|u) = p_{\pi^*}^{M^*}(v|u)$ para todos los pares (u, v).

Nótese la presencia del "apellido" estado a estado cuando se hace referencia a la equivalencia entre SSM. Esto es debido a que existe un concepto más general de equivalencia entre SSM que introduciremos más tarde (Definición 2.33) y, precisamente, se basa en la Definición 2.23.

Definición 2.24. Un estado s_i de una ISSM (M, π) es accesible si existe un par (u, v) tal que $\pi_i(v|u) \neq 0$. (u, v) puede ser el par (λ, λ).

Definición 2.25. Se dice que una ISSM es conexa si todos sus estados son accesibles.

Proposición 2.26. Si s_i es un estado accesible de (M, π), entonces existe un par (u, v) con $l(u, v) \leq |S| - 1$ tal que $\pi_i(v|u) \neq 0$.

Dem. Si s_i es accesible vía (u, v) con $l(u, v) = m$, entonces existe una secuencia de $m+1$ estados $s_1, \ldots, s_m$ tal que s_1 es un estado cuya componente asociada del vector π es no nula, la probabilidad de transición entre dos estados sucesivos es positiva para la secuencia de inputs-outputs dada y $s_m = s_i$.

Asumamos $m > |S| - 1$. Entonces el grafo que conecta la secuencia de estados contiene un bucle, es decir existen dos estados s_j, s_k de la secuencia tales que $s_j = s_k$. Basta iterar el proceso de sustituir (u, v) por el par (u', v') que elimina los símbolos correspondientes a los estados s_i con $j < i \leq k$ para obtener el par input-output buscado de longitud conjunta menor o igual que $|S| - 1$. □

El lema anterior implica que los estados accesibles de una ISSM se corresponden con aquellas coordenadas no nulas en al menos uno de los vectores $\pi(v|u)$ para los que $l(u, v) \leq |S| - 1$. Esto conduce a un método práctico para determinar los estados accesibles de una ISSM dada.

Teorema 2.27. Toda ISSM tiene una ISSM conexa equivalente.

Dem. Todo estado cuya coordenada de π asociada es no nula es accesible, luego si s_j no es accesible $\pi_j = 0$. Siguiendo lo anterior se construye el vector estocástico π' eliminando la j-ésima coordenada de π.

Nótese ahora que si s_j no es accesible y, para algún par (x, y) se tiene $a_{ij}(y|x) > 0$, entonces s_i tampoco es accesible. Considérese M' construida eliminando en todas las matrices las filas y columnas de M asociadas a estados no accesibles. Cabe decir que en este proceso se eliminan todas las filas correspondientes a las coordenadas no negativas de las columnas anteriormente eliminadas. Esto garantiza que M' sea una SSM.

Así pues, la ISSM conexa (M', π') es equivalente a (M, π). □

Definición 2.28. Sean A y B dos matrices cuadradas de orden r y s respectivamente. La matriz

$$A \dotplus B = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}, \tag{2.25}$$

de orden $r + s$ recibe el nombre de suma directa de A y B.

Corolario 2.29. La suma directa cumple las siguientes propiedades:

1. Si A y B son estocásticas, $A \dotplus B$ también lo es.

2. $(A_1 \dotplus B_1)(A_2 \dotplus B_2) = (A_1 A_2 \dotplus B_1 B_2)$ donde las matrices A_i tienen el mismo orden, así como las matrices B_i.

Definición 2.30. Sean $M = (S, X, Y, \{A(y|x)\})$ y $M = (S', X, Y, \{A'(y|x)\})$ dos SSM. La SSM $M \dotplus M' = (S \cup S', X, Y, \{A(y|x) \dotplus A'(y|x)\})$ recibe el nombre de suma directa de M y M'.

Teorema 2.31. Dos ISSM (M, π), (M', π') son equivalentes si y solo si son $(|S| + |S'| - 1)$-equivalentes.

Dem. La implicación $(\Rightarrow)$ es trivial. Sea M^* la suma directa de las dos SSM y sean ρ, ρ' vectores $(|S| + |S'|)$-dimensionales

$$\rho = (\pi_1, \ldots, \pi_{|S|}, 0, \ldots, 0), \quad \rho' = (0, \ldots, 0, \pi'_1, \ldots, \pi'_{|S'|}). \tag{2.26}$$

Entonces $p_\pi^M(v|u) = p_\rho^{M^*}(v|u)$ y $p_{\pi'}^{M'}(v|u) = p_{\rho'}^{M^*}(v|u)$. Asumiendo que las ISSM son $(|S|+|S'|-1)$-equivalentes, los cuatro miembros de las igualdades anteriores son iguales entre sí para todos los pares (u, v) tales que $l(u, v) \leq |S|+|S'|-1$. Entonces, siguiendo la Proposición 2.16, las distribuciones ρ, ρ' sobre M^* son equivalentes y, en consecuencia, también lo son las ISSM M, M‘. □

Definición 2.32. Sea $\mathscr{P}_n$ el conjunto de vectores estocásticos de dimensión n. $\mathscr{F}^M$ es el conjunto de funciones $\mathscr{F}^M = \{p_\pi^M : \pi \in \mathscr{P}_n\}$ de una SSM M.

Definición 2.33. Sean M, M' dos SSM. Se dice que M y M' son equivalentes si $\mathscr{F}^M \equiv \mathscr{F}^{M'}$. Es decir, si para cada distribución π existe una distribución π' (y viceversa) tales que (M, π) y (M', π') son ISSM equivalentes.

Proposición 2.34. Se verifican las siguientes propiedades:

1. $\mathscr{F}^M$ es cerrado bajo combinaciones convexas.
2. $\mathscr{F}^M$ es la envolvente convexa del conjunto de funciones $\mathscr{F}^M_S = \{p_{s_1}, \ldots, p_{s_n}\}$.
3. La equivalencia estado a estado de dos SSM implica la equivalencia de las mismas, mientras que el recíproco en general no es cierto.
4. Las dos condiciones siguientes son equivalentes para dos SSM M, M':
   a. $\mathscr{F}^M \equiv \mathscr{F}^{M'}$.
   b. $\mathscr{F}^M_S \subset \mathscr{F}^{M'}$ y $\mathscr{F}^{M'}_{S'} \subset \mathscr{F}^M$.

Dem.

1. Sea $\rho = (\rho_i) \in \mathscr{P}_n$, entonces
$$\sum_i \rho_i p_{\pi^i}(v|u) = \sum_i \rho_i \pi^i \eta(v|u) = \sum_i \rho_i \sum_j \pi^i_j \eta_j(v|u) =$$
$$\sum_j \sum_i \rho_i \pi^i_j \eta_j(v|u) = \sum_j \rho'_j \eta_j(v|u) = p_{\rho'}(v|u), \tag{2.27}$$
donde $\rho'_j = \sum_i \rho_i \pi^i_j$. Por tanto $\rho' = (\rho'_j) \in \mathscr{P}_n$.
2. Basta con observar que $p_\pi(v|u) = \sum \pi_i p_{s_i}(v|u)$. Se hará referencia a las funciones de la forma p_{s_i} como funciones extremales.
3. Nótese que $p_\pi(v|u) = \sum \pi_i p_{s_i}(v|u)$, luego que una SSM sea equivalente a otra estado a estado implica que para todo π existe un π' que satisface la definición de equivalencia entre SSM. En particular $\mathscr{F}^M_S = \mathscr{F}^{M'}_{S'}$.

   Por el contrario el recíproco no es cierto, pues que para todo π exista un π' y viceversa que satisfacen $\sum \pi_i p^M_{s_i}(v|u) = \sum \pi'_j p^{M'}_{s'_j}(v|u)$ no implica que los conjuntos $\{p^M_{s_i}\}$, $\{p^{M'}_{s'_j}\}$ sean iguales, por ejemplo podrían ser idénticos salvo porque uno tenga una combinación convexa de los demás vectores y el otro conjunto no.

4. La implicación $a \Rightarrow b$ se satisface trivialmente. Supongamos (b) cierto, entonces, dado que $\mathscr{F}_S^M \subset \mathscr{F}^{M'}$, y $\mathscr{F}^{M'}$ es la envolvente convexa de $\mathscr{F}_{S'}^{M'}$ se tiene que $p_{s_i}^M(v|u) = \sum a'_j p_{s'_j}^{M'}(v|u)$ para todos s_i y (u,v). Así pues, dado π distribución inicial de M, se tiene que para todo par (u,v):

$$p_{s_i}^M(v|u) = \sum_i \pi_i p_{s_i}^M(v|u) = \sum_j \sum_i \pi_i a'_{ij} p_{i,s'_j}^{M'}(v|u). \tag{2.28}$$

Puesto que todos los vectores a_i, π son estocásticos, el vector π' cuyas coordenadas son $\pi'_j = \sum_i \pi_i a'_{ij}$ es también estocástico y satisface las condiciones de la definición de equivalencia entre SSM. Un razonamiento análogo para la contención $\mathscr{F}_{S'}^{M'} \subset \mathscr{F}^M$ concluye la demostración demostrando que también se da el recíproco entre M' y M.

□

Teorema 2.35. Sea M una SSM de n estados tal que alguna fila de H^M es combinación convexa del resto. Entonces existe una SSM de $(n-1)$ estados M' equivalente a M.

Dem. Sean $h_1, \ldots, h_n$ las filas de H^M y asumamos que la fila $h_i = \sum_{j \neq i} a_j h_j$ con $(a_j) \in \mathscr{P}_n$ y $a_i = 0$. Entonces la envolvente convexa de las filas de H^M: $\mathrm{conv}(h_1, \ldots, h_n) = \mathrm{conv}(h_1, \ldots, h_{i-1}, h_{i+1}, h_n)$. Sea ahora ξ cualquier vector no nulo de cualquier matriz de M, entonces $\hat{\xi} = \xi / \sum \xi_i$ es un vector de $\mathscr{P}_n$ y $\sum \hat{\xi}_j h_j \in \mathrm{conv}(h_1, \ldots, h_n) = \mathrm{conv}(h_1, \ldots, h_{i-1}, h_{i+1}, h_n)$. Por tanto existe un vector $\rho \in \mathscr{P}_n$ con cordenada i-ésima nula tal que $\sum \hat{\xi}_j h_j = \sum_{j \neq i} \rho_j h_j$.

Así pues, $\xi H^M = (\sum \xi_i) \rho H^M$ y se pueden reemplazar todos los vectores ξ por los vectores $\sum \xi_i \rho$ correspondientes en todas las matrices de M. De esta forma se obtiene una SSM equivalente M'(Lema 2.18 y Proposición 2.34.3) cuyas columnas i-ésimas de todas las matrices son nulas. Basta ahora con considerar la ISSM (M', π) y, mediante un argumento análogo al anterior se puede encontrar un vector $\pi' \in \mathscr{P}_n$ con $\pi'_i = 0$ tal que (M', π') es equivalente. Y, dado que ahora el estado i-ésimo no es accesible, siguendo la demostración Teorema 2.27 se tiene que existe una ISSM (M^*, π^*) equivalente de $(n-1)$ estados. Esto garantiza la equivalencia entre las SSM M' y M^*, que conduce a la equivalencia entre M^* y M por transitividad. □

Definición 2.36. Se dice que una SSM M es mínima en estados si el conjunto de filas de H^M es convexamente independiente.

Corolario 2.37. Toda SSM M tiene una SSM mínima en estados equivalente M'.

Dem. Es consecuencia directa de la Definición 2.36 y el Teorema 2.35. □

Nótese que en la práctica, dada una SSM M, para encontrar una SSM M' mínima en estados se debe encontrar el subconjunto de vectores convexamente independientes de las filas de H^M. En particular, un vector h_i^M es combinación convexa de las otras filas de H^M si existe solución al problema de programación lineal siguiente:

Dada la matriz H^M, encontrar un vector $x = (x_1, \ldots, x_n)$ tal que $\sum_{j \neq i} x_j = 1$, $x_i = 0$ y $x_j \geq 0$ para todo j, que satisfaga $xH^M = h_i^M$ para cada fila h_i^M de H^M.

En cuanto a la implementación, existen múltiples métodos que resuelven este problema e incluso algunos que simplifican la búsqueda de algunas filas extremales de H^M.

2.2.2. Recubrimientos

Definición 2.38. Sean M y M^* dos SSM. Se dice que M recubre a M^* ($M \geq M^*$) si $\mathscr{F}^M \supseteq \mathscr{F}^{M^*}$.

Corolario 2.39. Dos SSM M, M^* son equivalentes si y solo si $M \geq M^*$ y $M \leq M^*$

Dem. Es consecuencia directa de la Proposición 2.34.4 y la Definición 2.38. □

Teorema 2.40. Las siguientes condiciones son equivalentes:

1. $M \geq M^*$.
2. Existe una matriz estocástica B tal que $B\eta^M(v|u) = \eta^{M^*}(v|u)$ para todos los pares (u, v) (i.e. $B[K^M] = [K^{M^*}]$).
3. Existe una matriz estocástica B que satisface la igualdad $BA^M(y|x)\eta^M(v|u) = A^{M^*}(y|x)B\eta^M(v|u)$ para todos los pares (u, v) y (x, y).
4. Existe una matriz estocástica B que satisface la igualdad $BA^M(y|x)H^M = A^{M^*}(y|x)B\,H^M$.

Dem. Se demostrarán secuencialmente las tres equivalencias:

(1) $\Leftrightarrow$ (2) Comencemos asumiendo (1) cierto y sea $\eta^{M^*}(v|u)$ un vector de K^{M^*}, entonces

$$\eta^{M^*}(v|u) = \begin{bmatrix} p^M_{s_1}(v|u) \\ \vdots \\ p^M_{s_{n^*}}(v|u) \end{bmatrix} = \begin{bmatrix} p^M_{\pi^1}(v|u) \\ \vdots \\ p^M_{\pi^{n^*}}(v|u) \end{bmatrix}$$
$$= \begin{bmatrix} \pi^1\eta^M(v|u) \\ \vdots \\ \pi^{n^*}\eta^M(v|u) \end{bmatrix} = \begin{bmatrix} \pi^1 \\ \vdots \\ \pi^{n^*} \end{bmatrix} \eta^M(v|u) = B\eta^M(v|u), \tag{2.29}$$

donde $p_{\pi^i}^M$ es la función en $\mathscr{F}^M$ igual a $p_{s_i}^{M^*}$ en $\mathscr{F}^{M^*}$ y B se construye como la matriz cuyas filas son los vectores π^i.

Asumamos ahora (2) cierto y sea $p_{\pi^*}^{M^*}$ una función cualquiera de F^{M^*}. Sea π la distribución $\pi = \pi^* B$, entonces, para todos los pares (u, v) se tiene que

$$p_{\pi^*}^{M^*} = \pi^* \eta^{M^*}(v|u) = \pi^* B \eta^M(v|u) = \pi \eta^M(v|u) = p_\pi^M(v|u), \tag{2.30}$$

y por tanto $\mathscr{F}^M \supseteq \mathscr{F}^{M^*}$.

(2) $\Leftrightarrow$ (3) Asumiendo (2) como cierto, basta con observar el siguiente diagrama para comprobar que se satisface (3).

$$\begin{array}{ccc} \eta^M(v|u) & \xrightarrow{A(y|x)} & \eta^M(yv|xu) \\ \downarrow B & & \downarrow B \\ \eta^{M^*}(v|u) & \xrightarrow{A^*(y|x)} & \eta^{M^*}(yv|xu) \end{array}$$

Recorrer el diagrama anterior desde $\eta^M(v|u)$ hasta $\eta^{M^*}(yv|xu)$ por el vértice superior derecho corresponde al miembro izquierdo de la igualdad, mientras que por el vértice inferior izquierdo corresponde al miembro derecho.

Supongamos (3) cierto. Se probará por inducción en $l(u, v)$ que (3) $\Rightarrow$ (2). Cuando la longitud de las palabras es nula la tesis se satisface trivialmente pues $\eta^M(\lambda, \lambda) = \eta^{M^*}(\lambda, \lambda) = \eta$. Asumamos ahora la hipótesis inductiva para el par (u, v), entonces

$$\begin{aligned} B\eta^M(yv|xu) = BA^M(y|x)\eta^M(v|u) = A^{M^*}(y|x)B\eta^M(v|u) = \\ A^{M^*}(y|x)\eta^{M^*}(v|u) = \eta^{M^*}(yv|xu). \end{aligned} \tag{2.31}$$

(3) $\Leftrightarrow$ (4) La implicación (3) $\Rightarrow$ (4) es trivial pues las columnas de H^M son vectores $\eta^M(v|u)$. Asimismo el recíproco es inmediato pues todo vector $\eta^M(v|u)$ es combinación lineal de las columnas de H^M. □

2.3. Minimización de estados en ISSM mediante recubrimientos

En la sección anterior hemos visto que dada M una SSM podemos encontrar otra equivalente reducida, es decir, sin filas iguales en H^M; así como una mínima en estados, todas las filas de H^M son convexamente independientes (más general). Ahora cabe preguntarse si podemos reducir aún más la SSM en el sentido de, suponiendo que ya es mínima en estados, encontrar una SSM que recubra a la original y tenga menos estados (problema I). De hecho, también es interesante considerar si se puede encontrar una SSM, también con menos estados, recubierta por la original (problema II). Sobre ambos problemas ni siquiera se sabe si son decidibles.

Por otro lado, el problema que se estudiará en esta sección (problema III) se puede reducir a los anteriores. El problema en cuestión es el siguiente:

Dada (M, π) *una ISSM, encontrar otra ISSM* (M^*, π^*) *tal que* $(M, \pi) \equiv (M^*, \pi^*)$ *y* $n^* < n$.

Definición 2.41. Dada (M, π) una ISSM, $G^{(M,\pi)}$ es la matriz cuyas filas, de la forma $\bar{\pi}(v, u)$ (ver Definición 2.7) para ciertos pares input-output (u, v), son linealmente independientes y cualquier vector de la forma $\bar{\pi}(v, u)$ es combinación lineal de las mismas.

Lema 2.42. Dada una ISSM (M, π) y una SSM M^*, existe un vector estocástico π^* con $(M, \pi) \equiv (M^*, \pi^*)$ si y solo si existe una matriz estocástica B^* tal que

$$B^*[K^{M^*}] = G^{(M,\pi)}[K^M]. \tag{2.32}$$

Dem. Supongamos que se cumple 2.32. Entonces, dado que la primera fila de $G^{(M,\pi)}$ es π $(= \bar{\pi}(\lambda, \lambda))$, el primer elemento de cada columna de la forma $G^{(M,\pi)}\eta^M(v|u)$ (miembro derecho de 2.32) es igual a $p_\pi^M(v|u)$. Su valor correspondiente en el miembro izquierdo es el primer elemento de la columna $B^*\eta^{M^*}(v|u)$ que, llamando π^* a la primera fila de B^*, es precisamente $p_{\pi^*}^{M^*}(v|u)$. Dado que esto se cumple para todas las columnas, se tiene que $(M, \pi) \equiv (M^*, \pi^*)$.

Asumamos ahora que existe un vector π^* tal que $(M, \pi) \equiv (M^*, \pi^*)$. Entonces $\pi^*[K^{M*}] = \pi[K^M]$ y por tanto $\pi^* A^{M^*}(v|u)[K^{M^*}] = \pi A^M(v|u)[K^M]$, pues las columnas de $A^M(v|u)[K^M]$ son un subconjunto de las de $[K^M]$ y se corresponden con las del miembro derecho de la igualdad, que cumplen la propiedad análoga. Ahora bien, cada fila $\bar{\pi}$ de $G^{(M,\pi)}$ es de la forma $a\pi A^M(v|u)$ siendo a una constante de normalización. Por tanto, para cada una de dichas filas, el vector $\bar{\pi}^* = a\pi^* A^*(v|u)$ cumple $\bar{\pi}^*[K^{M^*}] = \bar{\pi}[K^M]$ y la matriz B^* cuyas filas son los vectores $\bar{\pi}^*$ correspondientes a cada vector $\bar{\pi}$ de $G^{(M,\pi)}$ satisface la igualdad del enunciado. □

Teorema 2.43. Dada una ISSM (M, π) con n estados, existe una ISSM $(M^*, \pi^*) \equiv (M, \pi)$ con $n^* < n$ estados si y solo si existen una SSM M^* de n^* estados y una matriz estocástica B^* tales que

$$B^* A^{M^*}(y|x) H^{M^*} = \Delta^{(M,\pi)}(y|x) B^* H^{M^*}. \tag{2.33}$$

donde $\Delta^{(M,\pi)}(y|x)$ es una matriz que satisface $G^{(M,\pi)}A(y|x) = \Delta^{(M,\pi)}(y|x)G^{(M,\pi)}$.

Dem. Previamente a la demostración per se, es necesario hacer un pequeño desarrollo sobre la matriz $\Delta^{(M,\pi)}(y|x)$. Consideremos para ello la matriz $H^{(M,\pi)}$ que cumple que:

- Sus columnas son columnas de $[K^{(M,\pi)}]$.
- Sus columnas son base de las de $[K^{(M,\pi)}]$.
- Sus columnas son precisamente las primeras de $[K^{(M,\pi)}]$ que cumplen las dos condiciones anteriores.

Veamos que las columnas de $H^{(M,\pi)}$ son de la forma $\eta^{(M,\pi)}(v|u) = G^{(M,\pi)}\eta(v|u)$ con $l(u,v) \leq n-1$. Basta con notar que cualquier columna de $[K^{(M,\pi)}]$ es de la forma

$$G^{(M,\pi)}\eta(v|u) = G^{(M,\pi)}\sum a_i\eta_i = \sum a_i G^{(M,\pi)}\eta_i \tag{2.34}$$

donde η_i^M son las columnas de H^M y a_i son constantes. En consecuencia se puede construir la matriz $H^{(M,\pi)}$.

Dada ahora la matriz $H^{(M,\pi)}$ cuya columna i-ésima es $\eta^{(M,\pi)}(v|u)$ se define $H^{(M,\pi)}(y|x)$ como la matriz cuya columna i-ésima es $\eta^{(M,\pi)}(yv|xu)$ y, de manera análoga, se definen $H^{M^*}(y|x)$, $K^{(M,\pi)}(y|x)$ y $[K^{(M,\pi)}(y|x)]$. Entonces una matriz $\Delta^{(M,\pi)}(y|x)$ como la del enunciado del teorema satisface $G^{(M,\pi)}A(y|x)H^{M^*} = \Delta^{(M,\pi)}(y|x)G^{(M,\pi)}H^{M^*}$ y, por tanto,

$$G^{(M,\pi)}H^{M^*}(y|x) = \Delta^{(M,\pi)}(y|x)H^{(M,\pi)} \tag{2.35}$$

$$\Leftrightarrow \quad G^{(M,\pi)}[K^{M^*}(y|x)] = \Delta^{(M,\pi)}(y|x)[K^{(M,\pi)}] \tag{2.36}$$

$$\Leftrightarrow \quad [K^{(M,\pi)}(y|x)] = \Delta^{(M,\pi)}(y|x)[K^{(M,\pi)}], \tag{2.37}$$

por ser las columnas de $[K^M]$ combinación lineal de las de H^M, que a su vez están contenidas en las anteriores.

Por otro lado, las filas de $G^{(M,\pi)}$ son vectores probabilísticos, luego las filas de $G^{(M,\pi)}A(y|x)$ son vectores de la forma $\pi(v,u)$ (Definición 2.6). Ahora bien, atendiendo a la Definición 2.7, los vectores $\pi(v,u)$ son proporcionales a $\bar{\pi}(v,u)$, y todo vector $\bar{\pi}(v,u)$ es combinación lineal de las filas de $G^{(M,\pi)}$, así como lo son las filas de $\Delta^{(M,\pi)}(y|x)G^{(M,\pi)}$. Dado que las filas de $G^{(M,\pi)}$ son linealmente independientes, se deduce que existe una única matriz $\Delta^{(M,\pi)}(y|x)$ que satisface $G^{(M,\pi)}A(y|x) = \Delta^{(M,\pi)}(y|x)G^{(M,\pi)}$.

Ahora, dado el desarrollo anterior, es sencillo dar una prueba análoga a la del Teorema 2.40. Supongamos que, como enuncia el teorema, dada una ISSM (M,π) con n estados, existe una ISSM $(M^*,\pi^*) \equiv (M,\pi)$ con $n^* < n$ estados. Entonces, atendiendo al Lema 2.42 existe una matriz estocástica B^* de tamaño $n \times n^*$ con $n < n^*$ tal que

$$B^*[K^{M^*}] = [K^{(M,\pi)}], \tag{2.38}$$

donde $[K^{(M,\pi)}] = G^{(M,\pi)}[K^M]$. Y basta con considerar el siguiente diagrama para concluir que se cumple $\Delta^{(M,\pi)}(y|x)B^*\eta^{M^*}(v|u) = B^*A^{M^*}(y|x)\eta^{M^*}(v|u)$ para todos los pares (u,v) y (x,y).

$$\begin{tikzcd} \eta^{M^*}(v|u) \arrow[r, "A^{M^*}(y|x)"] \arrow[d, "B^*"] & \eta^{M^*}(yv|xu) \arrow[d, "B^*"] \\ \eta^{(M,\pi)}(v|u) \arrow[r, "\Delta(y|x)"] & \eta^{(M,\pi)}(yv|xu) \end{tikzcd}$$

En el diagrama anterior se ha suprimido el superíndice (M, π) de la matriz Δ únicamente por motivos de claridad visual.

Probemos el recíproco. Se asume que existe una matriz estocástica B^* de tamaño $n\times n^*$ con $n < n^*$ tal que $\Delta^{(M,\pi)}(y|x)B^*\eta^{M^*}(v|u) = B^*A^{M^*}(y|x)\eta^{M^*}(v|u)$ para todos los pares (u, v) y (x, y). Se busca demostrar que $B^*[K^{M^*}] = [K^{(M,\pi)}]$.

Procedamos por inducción en $l(u, v)$. Si $l(u, v) = 0$, es trivial que $B^*\eta^{M^*}(v|u) = G^{(M,\pi)}\eta^{M}(v|u)$ pues $\eta^{M^*}(v|u) = \eta^{M}(v|u) = \eta$. Supongamos la hipótesis inductiva cierta para el par (u, v), entonces

$$\begin{aligned} B^*\eta^{M^*}(yv|xu) &= B^*A^{M^*}(y|x)\eta^{M^*}(v|u) = \Delta^{(M,\pi)}(y|x)B^*\eta^{M^*}(v|u) \\ &= \Delta^{(M,\pi)}(y|x)\eta^{(M,\pi)}(v|u) = \eta^{(M,\pi)}(yv|xu). \end{aligned} \tag{2.39}$$

Ahora bien, recapitulando y recordando que las columnas de H^M son vectores $\eta^M(v|u)$ y todo vector $\eta^M(v|u)$ es combinación lineal de las columnas de H^M, se tiene que

$$\begin{aligned} & B^*[K^{M^*}] = [K^{(M,\pi)}] \\ \Leftrightarrow \quad & \Delta^{(M,\pi)}(y|x)B^*\eta^{M^*}(v|u) = B^*A^{M^*}(y|x)\eta^{M^*}(v|u) \\ \Leftrightarrow \quad & \Delta^{(M,\pi)}(y|x)B^*H^{M^*} = B^*A^{M^*}(y|x)H^{M^*}, \end{aligned} \tag{2.40}$$

para todos los pares (u, v) y (x, y). □

Este teorema reduce el problema III al problema I. La justificación escapa a los objetivos de este trabajo pero si se desea más información sobre el tema se remite al lector a la sección 5 del capítulo 1.B de [14]. A continuación se introducirán ciertas nociones relativas al problema II, que permitirán enunciar el Lema 2.49 y el consiguiente Teorema 2.50. Dicho teorema da condiciones suficientes para que el problema III sea decidible.

Recordemos que el problema II trataba de encontrar una máquina M^* recubierta por una máquina M dada, de menos estados. Esto, según el Teorema 2.40 equivale a

$$BA^M(y|x)H^M = A^{M^*}(y|x)BH^M. \tag{2.41}$$

Definición 2.44. Dadas M, H^M y ξ un vector fila subestocástico no nulo de dimensión adecuada, $h^M(\xi)$ es el punto $(\xi/\sum\xi_i)H^M$ en $\operatorname{conv}(h_1, \ldots, h_m)$.

Se dice que el vector ξ es simplicial si $h^M(\xi)$ es un punto frontera en el símplice $\operatorname{conv}(h_1, \ldots, h_m)$.

Definición 2.45. Si H^M es la matriz H asociada a la SSM M y sea A cualquier matriz no negativa de dimensión adecuada, entonces $h^M(A)$ es el conjunto de vectores no negativos de la forma $h^M(A_i)$ donde A_i es la fila i-ésima de A.

El teorema siguiente da una interpretación geométrica de la expresión 2.41, permitiendo así comprobar si una matriz B dada es solución del problema.

Teorema 2.46. Sea M una SSM. Existe otra SSM $M^* \leq M$ con $n^* < n$ estados si y solo si existe una matriz estocástica B de tamaño $n^* \times n$ tal que

$$\cup_{(y,x)} h^M(BA^M(y|x)) \subset \operatorname{conv} h^M(B). \tag{2.42}$$

Dada una matriz B como la anterior, se puede construir la matriz A^{M^*} buscada.

Dem. Supongamos que existe una SSM M^* como la del enunciado, entonces se satisface 2.41 para cierta matriz B del tamañano requerido. Sea $\xi = (\xi_1, \ldots, \xi_n)$ una fila no nula de $A^{M^*}(y|x)$ y $\zeta = (\zeta_1, \ldots, \zeta_n)$ la fila correspondiente de $BA^M(y|x)$. Por ser B estocástica preserva la norma 1, luego dado que $\zeta H^M = \xi BH^M$, se tiene que $\sum \zeta_i = \sum \xi_i$ y $(\zeta / \sum \zeta_i)H^M = (\xi / \sum \xi_i)BH^M$.

Concluyendo, $\xi / \sum \xi_i$ es un vector probabilístico y $(\xi / \sum \xi_i)BH^M$ es una combinación convexa de las filas de BH^M $((\xi / \sum \xi_i)BH^M \in \operatorname{conv} h(B))$. Por otro lado, $(\zeta / \sum \zeta_i)H = h(\zeta)$ por definición y por tanto $\cup_{(y,x)} h(BA^M(y|x)) \subset \operatorname{conv} h(B)$.

Asumamos ahora que existe B satisfaciendo las condiciones del teorema, entonces todo vector fila del miembro izquierdo de 2.41 es combinación convexa de los puntos de $h^M(B)$. Estas combinaciones convexas son de la forma $\alpha \zeta H^M$ siendo α una constante de normalización y ζ una fila de la matriz $BA^M(y|x)$ para cierto par (y, x). Así pues se tiene $\alpha \zeta H^M = \pi BH^M$, donde π es un vector estocástico.

Ahora bien, el teorema se satisface definiendo las matrices A^{M^*} como se sigue:

1. Si una fila en $BA^M(y|x)$ es nula, también lo será su correspondiente fila en $A^{M^*}(y|x)$.

2. Sea ζ un vector no nulo de $BA^M(y|x)$, entonces la fila correspondiente de $A^{M^*}(y|x)$ es $(1/\alpha)\pi$.

□

Corolario 2.47. Sea M una SSM de n estados y rango m. Sea $h_1^*, \ldots, h_{n^*}^*$ un conjunto de $n^* < n$ puntos en un espacio m-dimensional tal que

$$\cup_{(y,x)} h^M(A^M(y|x)) \subset \operatorname{conv}(h_1^*, \ldots, h_{n^*}^*) \subset \operatorname{conv}(h_1, \ldots, h_n), \tag{2.43}$$

entonces existe una SSM $M^* \leq M$ de n^* estados que se puede construir dados los puntos $h_1^*, \ldots, h_{n^*}^*$.

Dem. Sea B la matriz estocástica $n^* \times n$ tal que

$$BH^M = \begin{bmatrix} h_1^* \\ \vdots \\ h_{n^*}^* \end{bmatrix}, \tag{2.44}$$

que se puede encontrar sin problema, pues $(\mathrm{h}_1^*, \ldots, \mathrm{h}_{\mathrm{n}^*}^*) \subset (\mathrm{h}_1, \ldots, \mathrm{h}_{\mathrm{n}^*})$.

Ahora bien, para cualquier matriz estocástica B se cumple

$$\begin{aligned} \cup_{(y,x)} h^M(BA^M(y|x)) &\subset \operatorname{conv}\left(\cup_{(y,x)} h^M(A^M(y|x))\right) \\ &\subset \operatorname{conv}(h_1^*, \ldots, h_{n^*}^*) = \operatorname{conv} h^M(B), \end{aligned} \tag{2.45}$$

por definición de B. □

Corolario 2.48. Sea M una SSM de n estados. Si existe un subconjunto $(h_1^*, \ldots, h_{n^*}^*)$ del conjunto de puntos $(h_1, \ldots, h_{n^*})$ tal que

$$\cup_{(y,x)} h^M(A^M(y|x)) \subset \operatorname{conv}(h_1^*, \ldots, h_{n^*}^*), \tag{2.46}$$

entonces se puede construir una máquina $M^* \leq M$ de n^* estados.

En este punto estamos ya en situación de dar condiciones suficientes para la decidibilidad del problema III.

Lema 2.49. Sean $M \geq M^*$ dos SSM con $B[K^M] = [K^{M^*}]$ y sea π una distribución sobre M tal que $h^M(\pi) \in \operatorname{conv} h^M(B)$. Entonces existe una distribución π^* sobre M^* tal que $(M, \pi) \equiv (M^*, \pi^*)$.

Dem. Dado que $h^M(\pi) \in \operatorname{conv} h^M(B)$, existe π^* tal que $\pi H^M = \pi^* BH^M$. Puesto que π y $\pi^* B$ son vectores equivalentes para M (recordar Corolario 2.17), se cumple $\pi\eta^M(v|u) = \pi^* B\eta^M(v|u)$ para todos los pares (u, v). Finalmente se sigue que

$$p_\pi^M(v|u) = \pi\eta^M(v|u) = \pi^* B\eta^M(v|u) = \pi^*\eta^{M^*}(v|u) = p_{n^*}^{M^*}(v|u), \tag{2.47}$$

para todos los pares (u, v). □

Teorema 2.50. Sea (M, π) una ISSM de n estados. Existe una ISSM de $n^* < n$ estados $(M^*, \pi^*) \equiv (M, \pi)$ si se cumple una de las condiciones (1), (2) o (3) junto a la condición $(\star)$:

1. Existe una matriz estocástica B de tamaño $n^* \times n$ tal que

$$\cup_{y,x} h^M(BA^M(y|x)) \subset \operatorname{conv} h^M(B). \tag{2.48}$$

2. Existe un conjunto de n^* puntos $h_1^*, \ldots, h_{n^*}^*$ en un espacio m-dimensional ($m = \operatorname{rank} H^M$) tal que

$$\cup_{y,x} h^M(A^M(y|x)) \subset \operatorname{conv}(h_1^*, \ldots, h_{n^*}^*) \subset \operatorname{conv}(h_1, \ldots, h_n). \tag{2.49}$$

3. Existe un subconjunto $h_1^*, \ldots, h_{n^*}^*$ del conjunto de puntos $h_1, \ldots, h_n$ tal que

$$\cup_{y,x} h^M(A^M(y|x)) \subset \operatorname{conv}(h_1^*, \ldots, h_{n^*}^*). \tag{2.50}$$

⋆ $h^M(\pi) \in \operatorname{conv} h^M(B)$. Definiendo B en función de la condición satisfecha entre las anteriores: si se satisface (1), B se define como en dicha condición; si se satisface (2) o (3) se define B tal que

$$BH^M = \begin{bmatrix} h_1^* \\ \vdots \\ h_{n^*}^* \end{bmatrix}. \tag{2.51}$$

Dem. Este teorema es consecuencia de los Lemas 2.42 y 2.49, y los dos Corolarios 2.47 y 2.48. □

Si bien la noción de SSM isomorfa no se ha utilizado en este texto, conviene indicar que dos SSMs lo son si son iguales salvo permutaciones sobre sus estados.

3. Teoría de juegos evolutiva

3.1. Introducción

La teoría de juegos evolutiva estudia el comportamiento de grandes poblaciones de agentes en el marco de la teoría de juegos. Es decir se analiza un juego G de n jugadores considerando que hay una población $N >> n$ jugando a G. Este análisis se hace fundamentalmente desde el punto de vista poblacional, comprobando cómo el conjunto de individuos evoluciona en la elección de sus estrategias o cómo de óptimas y robustas son dichas estrategias, estableciendo así un paralelismo con la noción de EN.

Esta rama de la teoría de juegos tiene su origen en una serie de publicaciones del genetista y matemático John Maynard Smith ([19], [20] y [21]). De hecho, en estos artículos fundacionales se da el concepto de equilibrio básico en teoría de juegos evolutiva: la estrategia evolutivamente estable (EEE), que veremos más adelante. Otro de los fundamentos es la dinámica del replicador, que modela el estudio de la evolución poblacional considerando la reproducción y muerte de los individuos. Esto se hace a partir de la ecuación del replicador o ecuación replicadora introducida en [22] y nombrada así en [17].

A partir de ahora cuando se hable de poblaciones y evolución de las mismas se considerará que dicha población juega a un cierto juego entre todos los individuos. La composición de la población se ve modificada a lo largo del tiempo (por ejemplo iterativamente) haciendo que las estrategias más eficaces aumenten su frecuencia poblacional en detrimento de las menos beneficiosas.

Las definiciones siguientes, dadas por Maynard y Price, están formuladas para juegos bipersonales y simétricos. Un juego simétrico es aquel en el que todos los jugadores tienen conjuntos de estrategias puras idénticos y las transposiciones sobre el perfil de estrategias conducen a transposiciones sobre los mismos índices del vector de pagos.

Si bien la notación utilizada a continuación será la descrita en el Capítulo 1, cabe indicar que, para aumentar la claridad de lectura, se utilizarán los símbolos s y σ para referirse a estrategias y no a perfiles de estrategias como hasta ahora.

3.2. Conceptos y resultados fundamentales.

Definición 3.1. Sea G un juego bipersonal simétrico y $\sigma \in \Delta(S)$ una estrategia de G. σ es EEE si se satisface

1. $U_1(\sigma, \sigma) \geq U_1(\sigma', \sigma), \forall \sigma' \in \Delta(S)$

   y

2. Si $U_1(\sigma, \sigma) = U_1(\sigma', \sigma)$ entonces $U_1(\sigma, \sigma') > U_1(\sigma', \sigma'), \forall \sigma' \neq \sigma$.

La Definición 3.1 tiene su origen formal en [21] y es la empleada generalmente.

Esta definición formalizan la idea de que una estrategia σ es EEE cuando una población que juega dicha estrategia no puede ser invadida por una estrategia $\sigma' \neq \sigma$ empleada por una pequeña proporción de la población. De esta forma se refina el concepto de EN pues, si el perfil dado por todos los jugadores siguiendo σ no es EN, la existencia de un único individuo que juegue la estrategia que conduce al equilibrio provocará una invasión de la población.

Cabe señalar que la Definición 3.1 de EEE tiene un dominio muy restringido. Es por ello que en [23] se da una definición sobre estrategias puras generalizada a juegos bipersonales no simétricos. En este capítulo nos conformaremos con dar definiciones y resultados en la línea de la citada Definición 3.1, sin incluir demostraciones. Esto está motivado por dos factores: por un lado, las dimensiones de este trabajo se consideran ya suficientes y, por otro, tal y como se indicó previamente, se tiene previsto realizar una generalización de estos resultados en un futuro Trabajo Fin de Máster. En cualquier caso, todos los resultados irán acompañados de una referencia bibliográfica donde encontrar su demostración.

Dada una cierta población de individuos de n tipos distintos, la ecuación replicadora (ER) en su forma más general es

$$\dot{x}_i = x_i \left(f_i(x) - \phi(x) \right), \quad \phi(x) = \sum_{j=1}^{n} x_j f_j(x), \tag{3.1}$$

donde x_i es la proporción de individuos tipo i en la población; $x = (x_1, \ldots, x_n)$ es el vector estocástico que determina la distribución de los tipos; $f_i(x)$ es una función que denominaremos función de adaptación del replicador i y $\phi(x)$ es la media poblacional de adaptación. Queda implícitamente impuesto que $\sum x_j = 1$.

Definición 3.2. Se denomina punto estacionario de la ER a todo vector poblacional $x = (x_1, \ldots, x_n)$ tal que $x_i \left(f_i(x) - \phi(x) \right) = 0, \forall i \in \{1, 2, \ldots, n\}$.

- Se denomina punto asintóticamente estable o sumidero a todo punto estacionario x para el que existe $\epsilon > 0$ tal que si $||x - x'|| < \epsilon$, entonces la curva integral de (3.1) que pasa por x' en un instante t_0 y que denotaremos también por x' verifica $x'(t) \xrightarrow{t\to\infty} x$.

Cabe pensar que la función de adaptación de un individuo (o de un jugador con una estrategia predefinida) depende del resto de individuos de la población. Así pues, de ahora en adelante, si se considera que la ecuación (3.1) describe la evolución de un modelo de teoría de juegos, dada la distribución probabilística p sobre las estrategias del juego bipersonal simétrico G,

$$f_s\left(p(s)\right) = G(s|p), \tag{3.2}$$

es la función de adaptación de los jugadores que siguen la estrategia $s \in S$. Se considera también la función de adaptación de una distribución q como

$$G(q|p) = \sum_{s\in S} q(s)G(s|p). \tag{3.3}$$

Nótese que pese a que en la ER (ecuación (3.1)) se considera que la función de adaptación f_i puede ser diferente para cada individuo i, en la ecuación (3.2) la función G no depende de s, sino que lo toma como argumento. Esta sutileza es conceptualmente muy relevante pues la función de adaptación está definida sobre las estrategias, no sobre los jugadores. Por ello, en general, se asume que $G(s|p)$ es función de $u(s, s')$ para alguna estrategia s' del jugador rival. De hecho s' puede depender a su vez de la distribución p, que determina la frecuencia de cada estrategia entre todos los jugadores. En general, tal y como se indica en [22], se considera que $G(q|p)$ es, de hecho, la utilidad esperada de q visto como estrategia mixta frente a la estrategia mixta p del rival. Se asumirá dicha consideración también en este documento pues garantiza la equivalencia de las dos definiciones de EEE dadas.

En la siguiente definición, y de ahora en adelante, denominaremos estados a las distribuciones p sobre el conjunto de estrategias S del juego. Asimismo, dado que consideramos juegos finitos, se puede considerar que p es un vector estocástico con un número finito de elementos y se puede utilizar la notación $p(s) \equiv p_s$.

Definición 3.3. Dado un juego G bipersonal simétrico y una población (finita o infinita) de jugadores, decimos que el estado p es EEE si para cualquier otro estado q tal que $q \neq p$, considerando $\bar{p} = (1-\epsilon)p + \epsilon q$ se cumple

$$G(q|\bar{p}) < G(p|\bar{p}), \text{ para todo } \epsilon \text{ suficientemente pequeño.} \tag{3.4}$$

Esta definición de EEE depende de la población que estemos considerando así como de las funciones de adaptación. Por otro lado, la Definición 3.1 es inherente al propio juego G, lo cual permite relacionarlo con resultados anteriores de teoría de juegos.

Continuando con la comparación, esta nueva Definición 3.3, extraída de [22], considera únicamente estrategias puras, argumentándose hacia el final del documento citado que para considerar estrategias mixtas basta con tomar $p = (p^1, \ldots, p^k)$ sobre S siendo p^i la esperanza total de que se emplee la estrategia pura s_i. Es decir, computando tanto la proporción de jugadores tales que s_i se encuentra en el soporte de su estrategia σ, como las probabilidades que tiene cada uno de esos jugadores de emplearla. Esta última probabilidad viene dada precisamente por la coordenada i-ésima de la estrategia de cada uno de esos jugadores.

Cabe reseñar también que en [23] se da la versión de la definición anterior para juegos bipersonales no simétricos. En dicha definición se consideran únicamente estrategias puras, argumentando hacia el final del documento que para considerar estrategias mixtas basta con tomar $p = (p^1, \ldots, p^k)$ sobre S siendo p^i la esperanza de que se emplee la estrategia pura s_i en general. Es decir, computando tanto la proporción de jugadores tales que s_i se encuentra en el soporte de su estrategia σ, como la coordenada i-ésima de la estrategia de cada uno de esos jugadores, que representa las probabilidades que tiene tal jugador de emplearla.

De ahora en adelante se utilizará la denominación sistema evolutivo para referirse a la terna dada por una población, la función de adaptación que define la ER de dicha población y el juego subyacente.

Proposición 3.4. Dado un sistema evolutivo, todo EN del juego subyacente es un punto estacionario de la ER.

Ref. La demostración se pueden encontrar en [10]. □

Proposición 3.5. Dado un sistema evolutivo, un estado es EEE si y solo si es asintóticamente estable en ER.

Ref. La demostración se pueden encontrar en [10]. □

Nótese que todos los EN son puntos estacionarios de la ER, pero el recíproco no es necesariamente cierto. Ahora bien, todos los sumideros de la ecuación replicadora son EEE y, en consecuencia, EN.

En [22] se da una demostración de que los EEE son equilibrios estables de la ER en términos diferentes a los de la Proposición 3.5. En ambos casos se hacen ciertas suposiciones sobre la función de adaptación: en la demostración de la Proposición 3.5 se asume que es lineal, mientras que en [22] realmente se demuestra para funciones de adaptación suficientemente regulares. En cualquier caso, esto último se puede considerar cierto siempre en la práctica, pues están definidas sobre un compacto de $\mathbb{R}^n$ y por tanto, como consecuencia del teorema de aproximación de Weierstrass, se pueden aproximar con precisión arbitraria por funciones de clase C^∞.

3.3. Equilibrios refinados en teoría de juegos evolutiva.

Proposición 3.6. En un sistema evolutivo sobre un juego repetido no trivial no existen ningún perfil de estrategias puras o mixtas que sea EEE.

Ref. Se puede encontrar una demostración de lo anterior en el Apéndice A de [6]. □

Tal y como se indica en la Proposición anterior, la noción de EEE impone condiciones muy exigentes sobre el sistema para garantizar su existencia, en consecuencia, en teoría de juegos evolutiva se ha desarrollado otro concepto de equilibrio más débil.

Definición 3.7. Sea G un juego bipersonal simétrico y $\sigma \in \Delta(S)$ una estrategia de G. σ es una estrategia estable neutral (EEN) si se satisface

1. $U_1(\sigma, \sigma) \geq U_1(\sigma', \sigma), \forall \sigma' \in \Delta(S)$

   y

2. Si $U_1(\sigma, \sigma) = U_1(\sigma', \sigma)$ entonces $U_1(\sigma, \sigma') \geq U_1(\sigma', \sigma'), \forall \sigma' \neq \sigma$.

Esta sencilla relajación en la Definición 3.1 de EEE conduce a que pueda aparecer otra estrategia que desplace a la que determina el equilibrio, pero esta a su vez ha de ser desplazable por la original. Esto puede desembocar en una sucesión de invasiones de estrategias EEN que conduzcan a una estrategia que desplace definitivamente a las demás. La Definición 3.9, enunciada por primera vez en [6], caracteriza las estrategias que impiden que suceda una sucesión tal de invasiones.

Definición 3.8. Se definen respectivamente los conjuntos de estrategias peores, iguales y mejores que $\sigma \in \Delta(S)$ como sigue:

- $S_P(\sigma) = \{\mu \neq \sigma : U_1(\sigma, \sigma) \geq U_1(\mu, \sigma)$ y si $U_1(\sigma, \sigma) = U_1(\mu, \sigma)$, entonces $U_1(\sigma, \mu) > U_1(\mu, \mu)\}$.
- $S_I(\sigma) = \{\mu \neq \sigma : U_1(\sigma, \sigma) = U_1(\mu, \sigma)$ y $U_1(\sigma, \mu) = U_1(\mu, \mu)\}$.
- $S_M(\sigma) = \{\mu \neq \sigma : U_1(\sigma, \sigma) \leq U_1(\mu, \sigma)$ y si $U_1(\sigma, \sigma) = U_1(\mu, \sigma)$, entonces $U_1(\sigma, \mu) < U_1(\mu, \mu)\}$.

Definición 3.9. Sea G un juego bipersonal simétrico y $\sigma \in \Delta(S)$ una estrategia de G. σ es una estrategia robusta frente a invasiones indirectas (RFII) si se satisface

1. $S_M(\sigma) = \varnothing$

y

2. $\nexists \mu^1, \ldots, \mu^m$ con $m \geq 2$, tales que $\begin{cases} \mu^1 \in S_I(\sigma) \\ \mu^i \in S_I(\mu^{i-1}) \\ \mu^m \in S_M(\mu^{m-1}) \end{cases}$, $2 \leq i \leq n-1$.

De esta forma se obtiene una noción de equilibrio, sobre la que se puede comprobar rápidamente que es intermedia entre EEN y EEE. Así pues, el diagrama de Venn de la Figura 3.1 representa la relación entre los distintos equilibrios considerados hasta ahora en teoría de juegos evolutiva.

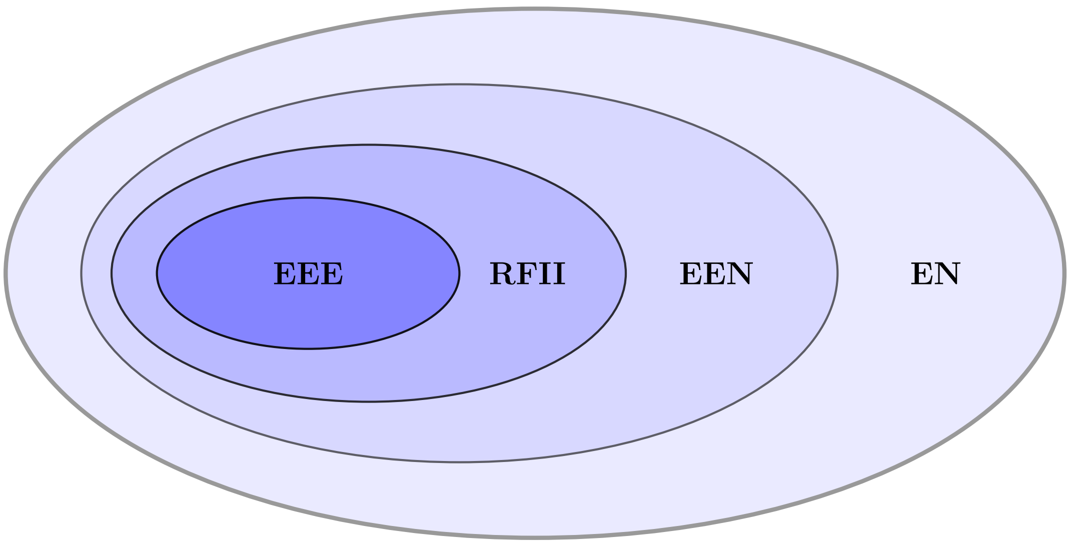

Figura 3.1: Figura reconstruida de [7], que muestra la relación entre los diversos equilibrios.

De nuevo en [6], se dan resultados que garantizan que una cierta estrategia no sea RFII tanto para el caso general como para el dilema del prisionero repetido infinitamente en particular.

Para concluir este capítulo, a continuación, se introducirá un breve desarrollo que concluye con un resultado que justifica el uso de autómatas para representar estrategias en el dilema del prisionero repetido infinitamente.

Sea $f : H_t \times S \times S \to \{0, 1\}$ la función dada por

$$f(h_t, s, s') = \begin{cases} 0 & \text{si } s(h_t) = s'(h_t) \\ 1 & \text{si } s(h_t) \neq s'(h_t) \end{cases} , \tag{3.5}$$

para todo $t \geq 1$.

Dado que estamos considerando juegos finitos, el conjunto de acciones tiene un cierto cardinal α finito. Tomando ahora $\delta \in (0,1)$ como factor de descuento, se define la distancia entre dos estrategias s y s' como

$$d(s,s') = (1-\delta)\sum_{t=1}^{\infty}\left(\frac{\delta}{k^2}\right)^{t-1}\sum_{h_t\in H_t} f(h_t,s,s'), \tag{3.6}$$

donde k es la cantidad de acciones posibles en cada nodo (se asume que esta cantidad es igual para todos los nodos).

Teorema 3.10. Dados un juego finito, bipersonal y simétrico G y la distancia d sobre $S \times S$, el conjunto de estrategias representables mediante autómatas es denso en el conjunto de estrategias de G.

Boceto de la demostración. Basta considerar el subconjunto $S_t \subset S$ compuesto por todas las estrategias de S en las que se juega la acción arbitraria a para todas las historias h_τ con $\tau > t$. Entonces el conjunto dado por la unión de todos los S_t para $t \geq 1$ es numerable y denso en S. Ahora bien, dado que todas las estrategias de S_t son representables por autómatas finitos para cualquier t, se concluye que, efectivamente, el conjunto de estrategias representables mediante autómatas es denso en S. □

Se puede encontrar una demostración detallada del teorema anterior en [6].

4. Implementación

4.1. Descripción general.

La implementación de la teoría desarrollada hasta ahora se hace siguiendo como modelo la publicación [12], que aplica el uso de autómatas finitos deterministas en el problema del dilema del prisionero repetido infinitamente. En esta memoria se generaliza el análisis a autómatas estocásticos y se actualiza la implementación. En el citado documento se codifican los autómatas mediante cadenas de bits, que limitan notablemente la generalización de la metodología a problemas más complejos en los que, por ejemplo, el juego considerado tenga más de dos acciones posibles o incluso más de dos jugadores. Estas cuestiones se abordarán con profundidad en un futuro Trabajo Fin de Máster, donde se espera dar una implementación mucho más general de esta metodología, aplicable a una amplia variedad de juegos facilitando la modificación de los parámetros que gobiernan la optimización y la descripción de las soluciones. Es por ello que este capítulo se limita a describir la implementación actual sin dar aplicaciones muy desarrolladas de la misma.

La metodología fundamentalmente consiste en crear una población aleatoria de autómatas que representen estrategias para el $\mathrm{DP}^{\infty}(\delta)$ y aplicar un algoritmo genético para obtener un EEN del juego.

En la implementación propuesta, los autómatas se codifican siguiendo el formalismo de Moore (Definición 2.8) de la siguiente manera:

- El atributo `.state0` del autómata contiene el vector estocástico π que determina la distribución probabilística inicial de los estados del autómata.

- El atributo `.actions` contiene un vector de $[0,1)^n$ donde n es el número de estados. El elemento i-ésimo del vector se corresponde con la esperanza matemática de la acción a llevar a cabo por el autómata en el estado i siendo 0 y 1 las acciones posibles (*Callar* y *Confesar* respectivamente).

- Los atributos `.trans0` y `.trans1` contienen las matrices estocásticas $A(\textit{Callar})$ y $A(\textit{Confesar})$.

Además se definen otros atributos del autómata de carácter operativo como el estado actual del autómata, que se utiliza cuando juega al $\mathrm{DP}^{\infty}(\delta)$. Dado que no tienen especial relevancia desde el punto de vista analítico, este tipo de detalles se pueden consultar en el Apéndice A.

4.2. Algoritmo genético.

Los algoritmos genéticos son métodos de optimización heurística que normalmente se aplican cuando la complejidad del problema impide determinar la solución de manera analítica. Se inspiran en el concepto de selección natural y el proceso evolutivo debidos a Charles Darwin.

Se suele considerar una población de posibles soluciones que evoluciona atendiendo a una función que describe su adaptación al medio. Las peores soluciones dentro de la población desaparecen iterativamente siendo reemplazadas por otras hipotéticamente mejores. Cada una de estas nuevas soluciones se obtiene a partir de un subconjunto de las supervivientes determinado siguiendo algún método particular de selección. Las soluciones seleccionadas se "reproducen" dando lugar a una solución hija que se define como función de ciertos parámetros que a su vez definen las soluciones progenitoras. Finalmente, se aplica algún tipo de mutación que añada cierta variabilidad a las nuevas soluciones y se comienza con la siguiente iteración. [9]

Para dar un carácter más general a estos algoritmos, cada uno de los individuos de la población se suele denominar célula o, como se hace en este documento, se denominarán en función del tipo de individuo poblacional que se utilice, en nuestro caso, autómatas.

Así pues, volviendo al caso que nos atañe, la figura 4.1 esquematiza el algoritmo genético utilizado. Se comienza creando una población aleatoria de N autómatas estocásticos de n estados en la representación de Moore. Estos autómatas comenzarán cada iteración jugando un torneo round robin del dilema del prisionero repetido "infinitamente". El formato round robin es lo que se conoce comúnmente como todos contra todos y, en nuestro caso, añadiremos también el encuentro espejo, es decir, cada autómata jugará también consigo mismo. Por otro lado, dado que no es posible simular el juego infinito, se realizan únicamente 150 repeticiones del juego de etapa. Siguiendo la implementación dada en [12], basta con considerar autómatas con n suficientemente pequeño en comparación con el número de etapas del juego para evitar que tengan memoria suficiente como para definir una estrategia que considere el final del juego como un evento alcanzable.

Tras la etapa de todos contra todos, se establece una clasificación de los autómatas ordenándolos de mayor a menor puntuación obtenida en el torneo. En la imple-

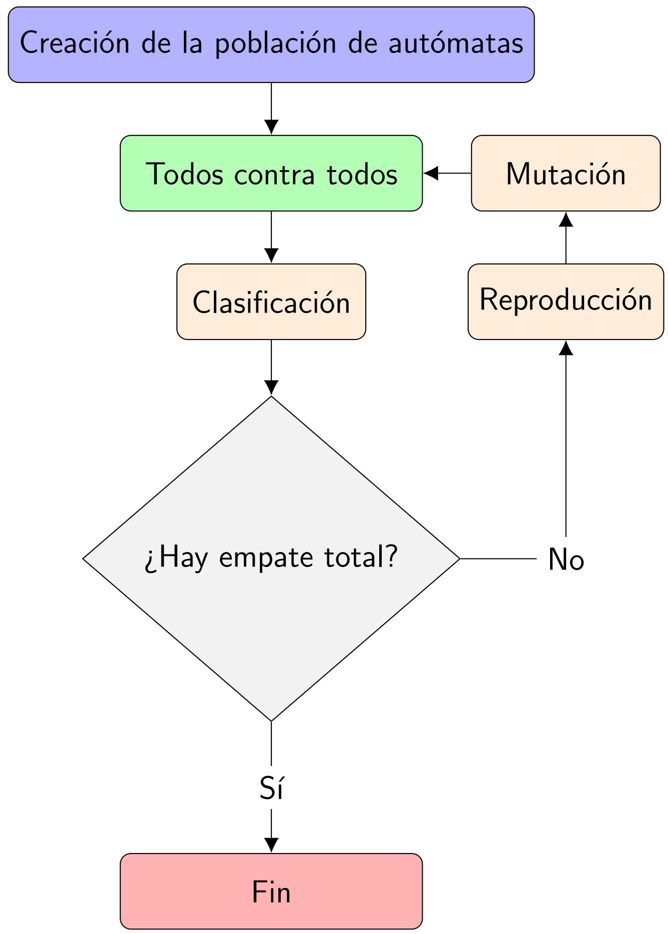

Figura 4.1: Diagrama de flujo del algoritmo genético.

mentación propuesta, por defecto se consideran poblaciones de $N = 30$ autómatas de los cuales sobreviven 20 por iteración. Así pues, cada uno de los últimos 10 clasificados se reemplaza por un nuevo autómata creado mediante una función `crossover` que toma como progenitores a dos autómatas elegidos aleatoriamente entre los 20 primeros considerando la distribución de probabilidad dada por sus resultados en la fase de round robin. Esta función se describe con detalle en la Sección 4.3.

Finalmente se añade una mutación a los nuevos autómatas y se vuelve a comenzar con el torneo round robin. El proceso de mutación consiste en modificar con probabilidad 0.005 cada uno de los elementos de la matriz de transición, de la distribución inicial del autómata y de las probabilidades de llevar a cabo una acción determinada en cada estado. Aquellos que se modifiquen se sustituyen por un valor aleatorio en el intervalo $[0, 1)$ con distribución de probabilidad uniforme. Finalmente, dado que muchos de los elementos citados están sujetos a la restricción de formar parte de un vector estocástico, se realizan las transformaciones pertinentes para que no se viole dicha restricción.

En la implementación dada se incluyen también dos bloques de control. El primero aparece en la figura 4.1 y detiene el algoritmo cuando todos los autómatas de la población han obtenido los mismos resultados. El segundo consiste en forzar manualmente a que el algoritmo no se repita más de un numero variable de iteraciones, que en este caso se ha fijado en 40. Esto evita que se entre en un bucle infinito si nunca se llegan a homogeneizar los resultados del torneo round robin.

El código que pone en práctica esta implementación es notablemente flexible ya que permite cambiar multitud de parámetros tales como el número de estados de los autómatas considerados, el tamaño de la población, la cantidad de individuos reemplazados por iteración, el juego utilizado, etc. Asímismo es fácil redefinir las funciones que gobiernan la reproducción y la mutación de los autómatas.

Además se incluye una traducción a Python de la implementación dada en [12] y la implementación de autómatas puramente aleatorios, cuyas transiciones no dependen de las jugadas del rival. Estos otros dos tipos de autómata pueden utilizarse como test, población de control o comprobación de resultados.

Se puede consultar el código para comprobar todo lo anterior en el Apéndice A.

4.3. Función `crossover`.

La literatura que relaciona el dilema del prisionero con autómatas finitos es amplia. Tanto en el ya visto caso del dilema repetido infinitamente con autómatas deterministas [12], como en el juego no repetido con autómatas estocásticos (se deja como referencia una colección de algunos de ellos: [2], [8], [18]). Durante la documentación bibliográfica de este trabajo se ha encontrado incluso una formulación cuántica [1].

Por el contrario, no se ha podido encontrar ninguna publicación que aborde el dilema del prisionero repetido infinitamente con autómatas estocásticos. Esto podría estar relacionado con el hecho de que tampoco se ha encontrado ninguna publicación en la que se utilice una función `crossover` que se pueda aplicar a matrices o vectores estocásticos dentro de un algoritmo genético. De hecho esta búsqueda ha sido infructuosa en general, no sólo considerando el dilema del prisionero como marco de aplicación. Es por ello que esta sección está dedicada a describir el problema de encontrar una función apropiada y a exponer la solución propuesta por el autor del trabajo.

El problema radica en determinar una función que muestre un comportamiento adecuado a la hora de combinar autómatas. Recordemos que la implementación utilizada es la descrita en la Sección 4.1, luego la función `crossover` debe ser capaz de tomar dos autómatas $A1, A2$ y mezclar sus vectores estocásticos `state0`, los vectores `actions` y las dos matrices estocásticas `trans0` y `trans1` para obtener un nuevo autómata.

Dado que se está siguiendo la implementación dada en [12] es razonable aplicar la función utilizada en el artículo. Basta con ordenar todos los elementos de los vectores y matrices indicados en el párrafo anterior, deteminar un elemento pivote p, una cantidad de elementos l y construir dos vectores mediante el siguiente algoritmo:

1. Crear una copia $A1'$ del vector que representa $A1$ y una copia $A2'$ del vector que representa $A2$.

2. Sustituir el p-ésimo elemento de $A1'$ y los $l-1$ elementos situados a su derecha por los correspondientes elementos de $A2$. En caso de que sólo haya $l_1 < l-1$ elementos a la derecha de p, sustituir esos l_1 elementos y los $l-1-l_1$ primeros del vector.

3. Sustituir el p-ésimo elemento de $A2'$ y los $l-1$ elementos situados a su derecha por los correspondientes elementos de $A1$. En caso de que sólo haya $l_2 < l-1$ elementos a la derecha de p, sustituir esos l_2 elementos y los $l-1-l_2$ primeros del vector.

Finalmente basta con volver a construir los elementos pertinentes de los autómatas a partir del vector. Con este método obtenemos dos autómatas en lugar de uno, lo cual permite sustituir dos individuos de la población en el algoritmo genético con una sola aplicación de la función `crossover`.

Esta propuesta no se ha considerado aceptable por diversos motivos que se describirán a continuación sin profundizar en los detalles. El primero de ellos es que computacionalmente no es especialmente eficiente, pues conlleva determinar dos números aleatorios, lo cual es notoriamente costoso. Además, es altamente probable que esta transformación utilice como pivote un elemento que forme parte de un vector estocástico, provocando que el vector correspondiente de los autómatas "hijos" no sea estocástico. Esto se puede solucionar dividiendo cada coordenada del vector entre la suma de todas ellas. Cabe indicar que esto también ocurre en la última coordenada intercambiada (a partir de ahora denominada contrapivote), no solo en el pivote.

Si bien la problemática descrita en el párrafo anterior es asumible, existe una consideración que impide tomar el mecanismo anterior como `crossover`. Supóngase que el vector `state0` óptimo es determinista, es decir, todas sus coordenadas son nulas a excepción de la coordenada i-ésima que tiene valor 1. Si partimos de progenitores puramente estocásticos con `state0` tal que su coordenada i-ésima no es significativamente mayor que las demás, es extremadamente improbable que aplicaciones sucesivas de la función `crossover` converjan hacia el óptimo. Esto se debe a que el pivote o el contrapivote deben encontrarse en `state0` y, además, el intercambio debe provocar que los vectores hijo tiendan a representar distribuciones probabilísticas cada vez más apuntadas. Una demostración rigurosa de lo anterior escapa a los objetivos de este trabajo, pero basta con simular este procedimiento para obtener una conclusión heurística que lo avale.

A favor de esta función `crossover` cabe destacar que es fácilmente interpretable, pues los estados, acciones y distribución inicial de los autómatas hijo son los de uno de los padres excepto para, como mucho, la distribución inicial y uno o dos estados. Además, tal y como se muestra en el artículo original, para autómatas deterministas favorece una rápida convergencia del algoritmo genético.

Este breve análisis permite deducir un conjunto de "buenas propiedades" que debe presentar la función buscada:

1. Se debe poder alcanzar el óptimo partiendo de cualquier población inicial de autómatas.

2. Conviene que sea computacionalmente eficiente, pues ha de repetirse en cada etapa del algoritmo genético tantas veces como autómatas deseemos sustituir.

3. Conviene que sea interpretable, pues toda la metodología descrita en este trabajo se ha desarrollado buscando facilitar la extracción de información del proceso de optimización.

Nótese que en la práctica, basta con encontrar una función `crossover` que se aplique a vectores estocásticos, pues tanto `state0` como las filas de `trans0` y `trans1` lo son. El vector `actions`, como veremos, es más fácil de manipular.

En el proceso de elección de la función `crossover` se han considerado transformaciones como la media aritmética de las coordenadas y la media geométrica.

La media geométrica requiere una normalización del vector hijo, entendiendo normalización como división de cada coordenada por la suma de todas ellas. Esto se debe a que la suma de los productos coordenada a coordenada es igual a 1. Sin entrar en si esta transformación posee las propiedades (1) y (2), la normalización de los vectores, previa multiplicación coordenada a coordenada, impide interpretar con facilidad el vector resultante. En consecuencia, se descartó su uso.

La media aritmética presenta las propiedades (2) y (3), pues es una transformación intuitiva y tanto para Matlab como para Python tiene un bajo coste computacional, además, el vector resultante es un vector normalizado. Por el contrario no cumple con la propiedad (1), pues dado un conjunto inicial de vectores estocásticos vistos como puntos de un espacio afín, aplicar iterativamente esta transformación no permite alcanzar vectores que determinen puntos no pertenecientes a la envolvente convexa de los iniciales.

Así pues, se parte de la media aritmética para construir la función `crossover` buscada. Para describirlo se hará uso de un ejemplo ilustrativo.

Considérense autómatas de 3 estados. En este caso, los vectores estocásticos que se deben tratar son tridimensionales $(x, y, z) \in \mathbb{R}^3$ y cumplen $x + y + z = 1$, $x \geq 0, y \geq 0, z \geq 0$.

Definición 4.1. $\Delta_n = \{(x_1, \ldots, x_n) : \sum_{i=1}^{n} x_i = 1 \text{ y } x_i \geq 0, \forall i = 1, \ldots, n\}$ es el conjunto de vectores estocásticos de $\mathbb{R}^n$ para $n \geq 2$.

La Figura 4.2 representa Δ_3, junto al punto $M = (0{,}14, 0{,}21, 0{,}65)$ y el punto $P = (0{,}46, 0{,}34, 0{,}20)$, que son vectores estocásticos.

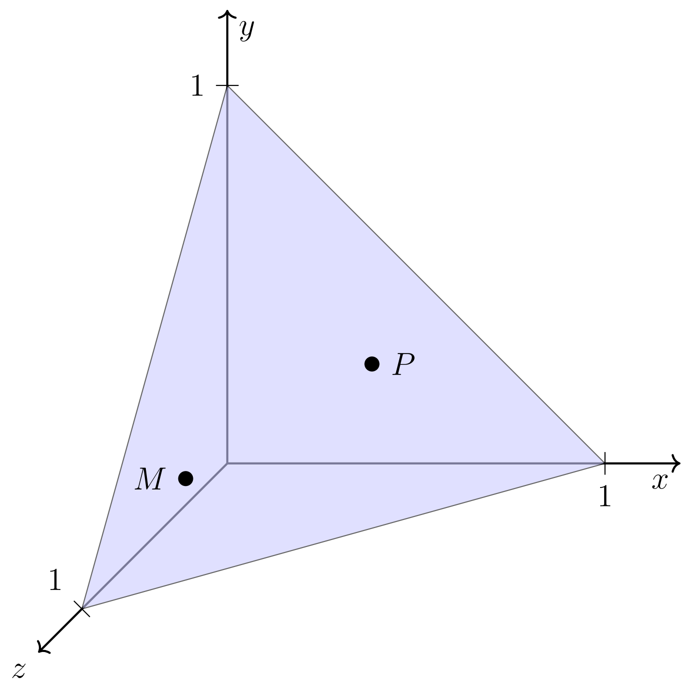

Figura 4.2: Representación del espacio de vectores estocásticos.

M representa la media de los dos vectores estocásticos progenitores y P el vector estocástico perdedor, que ha de ser sustituido por el resultado de la función `crossover`. Para conseguir que dicha función satisfaga la propiedad (1) se puede intentar considerar como vector hijo al punto simétrico de P con respecto de M, es decir $T = M + \overrightarrow{PM}$. El problema de esta consideración es que, tanto en el ejemplo presentado como en gran parte de los casos, el vector $T = (-0{,}18, 0{,}08, 1{,}10)$ no es estocástico. Esto aparece representado gráficamente en la Figura 4.3.

Para escapar de este atolladero se propone una solución inspirada en la Teoría de la Relatividad Especial (consultar [16]). Según la mencionada teoría, existe una ve-

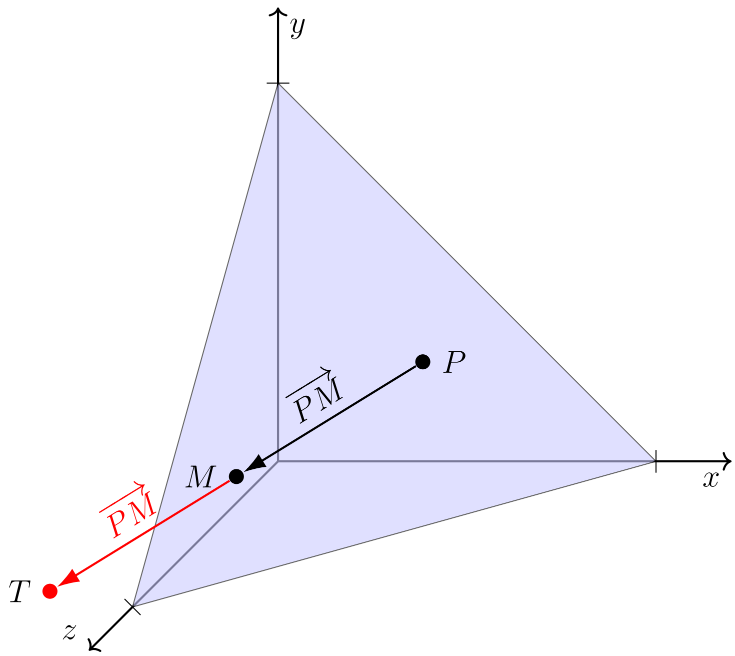

Figura 4.3: Propuesta inválida de `crossover`.

locidad máxima universal que no puede ser rebasada por ningún cuerpo: la velocidad de la luz en el vacío c. Este comportamiento viene modelado por la Transformación de Lorentz de velocidades, que describe la adición de las velocidades paralelas $\vec{u}, \vec{v}$ como

$$w = \frac{u+v}{1+uv/c}, \tag{4.1}$$

donde u y v representan los módulos de las velocidades.

Es sencillo comprobar que $\forall u, v$ pertenecientes al intervalo $[0, c]$, $w \in [0, c]$. En particular, si $u = c$ o $v = c$, entonces $w = c$.

Volviendo al problema planteado, se puede considerar la ecuación paramétrica de la recta que pasa por P y por M, que tendrá dos puntos de corte con la frontera de Δ_3. Sea A el punto de corte que dista menos de P que de M, Ω el punto de corte que dista menos de M que de P y sea $c = |\overrightarrow{\mathrm{A}\Omega}|$, entonces el vector hijo de la función `crossover` es

$$H = \mathrm{A} + w\,\frac{\overrightarrow{\mathrm{A}\Omega}}{c} = (0{,}06, 0{,}17, 0{,}77), \tag{4.2}$$

donde

$$w = \frac{|\overrightarrow{\mathrm{A}M}| + |\overrightarrow{PM}|}{1 + |\overrightarrow{\mathrm{A}M}||\overrightarrow{PM}|/c}. \tag{4.3}$$

Cabe aclarar que, pese a que se haya denominado normalizar a dividir entre la norma de la suma de las coordenadas de un vector, la notación $|\cdot|$ hace referencia a la noción usual de módulo de un vector.

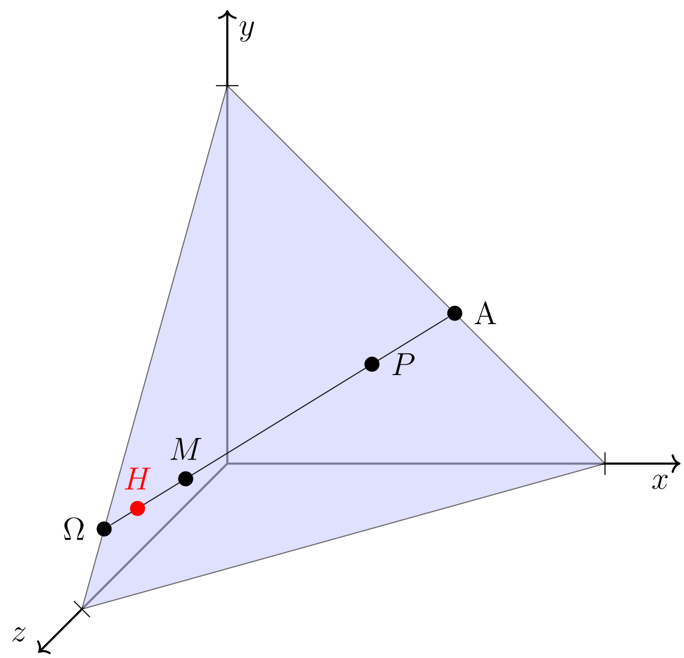

Figura 4.4: Propuesta inválida de `crossover` con corrección relativista.

Queda por salvar un último escollo. Puede darse que el autómata que se debe eliminar en el algoritmo genético P haya obtenido una puntuación mejor que la media en el torneo round robin. De hecho, esta podría ser igual que la de los dos progenitores M_1, M_2 del autómata que le sustituirá A y puede ser, incluso, que el óptimo O se encuentre en la envolvente conexa de los tres. En ese caso, calculando $A = H$ como antes, se podría estar alejando A de O (ver Figura 4.5).

Así pues, para facilitar la convergencia en estos casos y, en general, dar un peso a la influencia del autómata eliminado sobre el que le sustituirá, se considera como autómata imagen de la función `crossover` a aquel cuyos vectores estocásticos vienen determinados por

$$A = M + d\,\overrightarrow{MH}, \text{ con } d = \begin{cases} 1 & \text{si } 1 \leq \frac{\bar{w}-w_P}{\sigma} \\ \frac{\bar{w}-w_P}{\sigma} & \text{si } 0 \leq \frac{\bar{w}-w_P}{\sigma} < 1 \\ \frac{\bar{w}-w_P}{3\sigma} & \text{si } -1 < \frac{\bar{w}-w_P}{\sigma} < 0 \\ -1/3 & \text{si } \frac{\bar{w}-w_P}{\sigma} \leq -1 \end{cases}, \tag{4.4}$$

donde w_P es la puntuación obtenida por el autómata que se debe eliminar en la fase

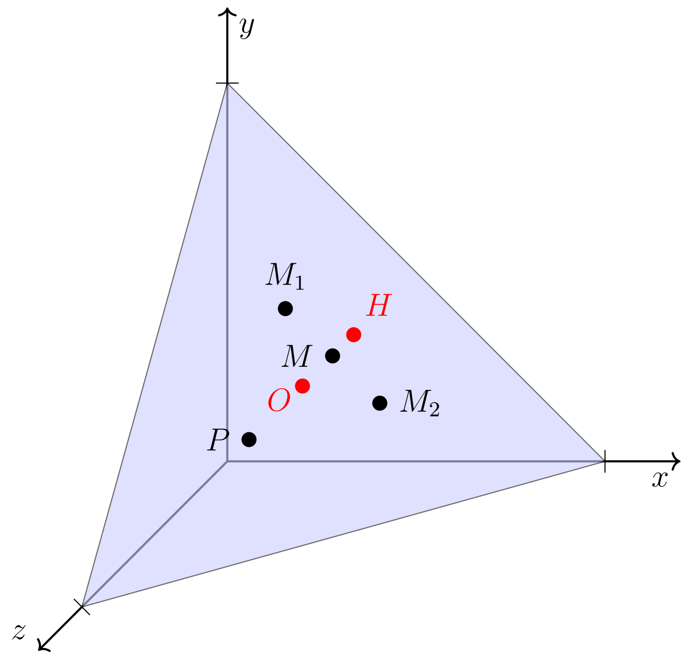

Figura 4.5: Ilustración de por qué la corrección relativista no es suficiente.

round robin, $\bar{w}$ es la media muestral de las puntuaciones de todos los autómatas y σ su desviación estándar.

La ecuación (4.4) está diseñada tomando como parámetro fundamental la cantidad de desviaciones estándar que dista el resultado w_P de la media. Si es menor que la media y dista menos de una desviación estándar de ella, este parámetro determina el peso que tiene el autómata eliminado (con la correspondiente corrección relativista) sobre el autómata que lo sustituye. Dado que dar un peso d superior a 1 daría al traste con la corrección, se toma $d = 1$ si w_P dista más de una desviación estándar de la media.

En caso de que w_P sea superior a la media no tendría sentido que contribuyera negativamente en la determinación del nuevo autómata. Pero, dado que se ha clasificado peor que los dos autómatas progenitores, es razonable reducir su impacto dividiendo por un factor 3. Este valor es considerablemente arbitrario y sería interesante un futuro estudio del mismo. Por lo demás, el razonamiento es análogo al caso en que $w_P \geq 0$.

De ahora en adelante, se denotará a la cantidad de desviaciones estándar que dista w_P de la media por

$$\lambda_P = \frac{\bar{w} - w_P}{\sigma}. \tag{4.5}$$

Esta función es una propuesta de solución propia del autor al problema de mezclar autómatas estocásticos para obtener otros nuevos. Dado que no se ha encontrado referencia alguna sobre algo semejante a lo expuesto en esta sección, la demostración de si la función `crossover` está dotada de las buenas propiedades que se esperan de ella requiere de un trabajo profundo y laborioso que escapa a lo esperable de un Trabajo Fin de Grado. Es por ello que se concluye esta sección formalizando la función para autómatas generales de n estados y se conjetura que está dotada de las propiedades (1) y (2). Además, las explicaciones y ejemplos previos se han dado, precisamente, en aras de verificar que se satisface la propiedad (3).

Definición 4.2. Sean $M_1, M_2, P \in \Delta_n$ y $\lambda \in \mathbb{R}$. Se define la función $\kappa : \Delta_n \times \Delta_n \times \Delta_n \times \mathbb{R} \longrightarrow \Delta_n$ como

$$\kappa(M_1, M_2, P, \lambda) = \begin{cases} M & \text{si } M \in \partial\Delta_n \\ M + d_\lambda \left(\mathrm{A} + \frac{|\overrightarrow{\mathrm{A}M}| + |\overrightarrow{PM}|}{|\overrightarrow{\mathrm{A}\Omega}| + |\overrightarrow{\mathrm{A}M}||\overrightarrow{PM}|} \overrightarrow{\mathrm{A}\Omega} - M \right) & \text{si } M \notin \partial\Delta_n \end{cases}, \tag{4.6}$$

donde $\partial\Delta_n$ representa la frontera de Δ_n, $M = (M_1 + M_2)/2$ y

$$d_\lambda = \begin{cases} 1 & \text{si } 1 \leq \lambda \\ \lambda & \text{si } 0 \leq \lambda < 1 \\ \lambda/3 & \text{si } -1 < \lambda < 0 \\ -1/3 & \text{si } \lambda \leq -1 \end{cases}. \tag{4.7}$$

Definición 4.3. Sean $a_1, a_2, a_P \in [0, 1]$ y $\lambda \in \mathbb{R}$. Se define la función $c : [0, 1] \times [0, 1] \times [0, 1] \times \mathbb{R} \longrightarrow [0, 1]$ como

$$c(a_1, a_2, a_P, \lambda) = \begin{cases} m + d_\lambda \left(\frac{2m - a_P}{1 + m^2 - m a_P} - m \right) & \text{si } m \geq a_p \\ m + d_\lambda \left(1 - \frac{1 - 2m + a_P}{1 + a_P - m - m a_P + m^2} - m \right) & \text{si } m \leq a_p \end{cases}, \tag{4.8}$$

donde $m = (a_1 + a_2)/2$ y

$$d_\lambda = \begin{cases} 1 & \text{si } 1 \leq \lambda \\ \lambda & \text{si } 0 \leq \lambda < 1 \\ \lambda/3 & \text{si } -1 < \lambda < 0 \\ -1/3 & \text{si } \lambda \leq -1 \end{cases}. \tag{4.9}$$

En la definición siguiente se asume que la palabra autómata hace referencia a autómatas implementados como se ha descrito en este capítulo y se considera que para un cierto autómata α, las expresiones α`.state0`, α`.actions`, α`.trans0` y α`.trans1` denotan sus atributos correspondientes. Por otro lado, se asume que la función K es la función que aplica κ a las filas de las matrices que toma como argumento y C aquella que aplica c a los elementos de los vectores que toma como argumento.

Definición 4.4. Sean A_1, A_2, A_P autómatas secuenciales estocásticos y λ_P la cantidad de desviaciones estándar que dista la puntuación de A_P de la media. Se denomina crossover de A_1 y A_2 en A_P a la función

$$crossover : (A_1, A_2, A_P, \lambda_P) \longmapsto A, \tag{4.10}$$

donde:

- A.`state0`$= \kappa(A_1$.`state0`,A_2.`state0`,A_P.`state0`,$\lambda_P)$.
- A.`trans0`$= K(A_1$.`trans0`,A_2.`trans0`,A_P.`trans0`,$\lambda_P)$.
- A.`trans1`$= K(A_1$.`trans1`,A_2.`trans1`,A_P.`trans1`,$\lambda_P)$.
- A.`actions`$= C(A_1$.`actions`,A_2.`actions`,A_P.`actions`,$\lambda_P)$.

Conjeturas:

1. La función *crossover* de la Definición 4.4 tiene complejidad computacional $\mathcal{O}(n^2)$.
2. Sea A un autómata secuencial estocástico y, dado un juego G, sea $\{\mathcal{A}(G)_k\}_{k=1}^{\infty}$ la sucesión de poblaciones $\mathcal{A}(G)_k$ determinada por el algoritmo genético descrito en esta sección con la función *crossover* de la Definición 4.4 y sin considerar mutaciones. Existen un juego G_A y una población inicial $\mathcal{A}_1$ tales que la sucesión $\{\mathcal{A}(G_A)_k\}_{k=1}^{\infty}$ con $\mathcal{A}(G_A)_1 = \mathcal{A}_1$ converge hacia A.

Es importante reseñar que esta función `crossover` tiene una “buena propiedad” que no habíamos estudiado previamente: no depende de variables aleatorias. Esto es muy enriquecedor, ya que esta función determina una dinámica sobre la población de autómatas de manera unívoca. De esta forma se puede estudiar la ecuación del replicante, así como determinar sumideros o describir la órbita de una población inicial dada. Este tratamiento también conduce a dirimir si para cada punto de Δ_n existe un juego tal que el punto es alcanzable mediante el algoritmo genético, o si puede existir una sucesión de poblaciones que converja hacia dicho punto.

Todo este estudio junto a la demostración de las conjeturas planteadas constituye un posible Trabajo Fin de Máster que se contempla como continuación de este.

4.4. Otras disquisiciones interesantes.

Se considera relevante reseñar la importancia de la implementación en el tratamiento del problema. Más allá de las cuestiones generales, por ejemplo de optimización, en las que influye la implementación dada, cabe incidir en las implicaciones conceptuales que se dan en el caso tratado.

Un autómata se puede representar de múltiples formas diferentes según la implementación descrita. Por ejemplo, considérese el autómata resultado de trasponer dos estados y las columnas correspondientes en las matrices `trans0` y `trans1` así como las coordenadas correspondientes de `state0` para un autómata dado. El nuevo autómata tendrá un comportamiento idéntico al original, pero se comportará de manera claramente diferente al "relacionarse" con otros mediante la función `crossover`.

De hecho, puede ocurrir que existan estados inaccesibles para un autómata, que no tendrán influencia en su comportamiento durante el juego, pero sí la tendrán durante el proceso evolutivo. De hecho, puede ocurrir que esos autómatas representen estrategias que no son EEN pero que se comportan como tal, pues al aparecer una población invasora, la función `crossover` puede hacer que los estados inaccesibles comiencen a serlo y ese otro autómata expulse la invasión. Una posterior optimización de esta población de autómatas una vez expulsados los invasores puede conducir de nuevo a la población inicial aparentando que la estrategia considerada es un EEN. Es decir, la implementación influye directamente en la resolución del problema.

A. Código Python

```python
# El código está realizado con vistas a ser adaptable a casos más generales
# sin tener que realizar modificaciones sobre el mismo. En particular se
# posibilita la inclusión de redes neuronales como jugadores.


# Importación de librerias

import numpy as np
import itertools,copy,time,random



# DECLARACIONES

# Esta clase juega la función de determinar las acciones de cada jugador,
# puede ser un autómata de moore, una red neuronal...
class Strategy():
    # Atributos y métodos obligatorios en cualquier estrategia
    def __init__(self):
        self.score = 0

    # Este método debe existir pues hay jugadores como los autómatas que deben
    # actualizarse en cada etapa del juego y hay otros como las redes que no
    # tienen que actualizarse. Dado que hay algunos que sí, en todos los juegos
```

```python
    # ha de aparecer una llamada por jugador al método update, el cual será un
    # método vacío en los casos en los que el jugador no deba actualizarse.
    def update(self):
        pass



# Función normalizadora de los filas (a partir del segundo término) de matrices
def _normVect(vect):
    norm=np.sum(vect)
    if norm==0 or norm==1:
        return vect
    vect/=norm
    return vect

# Función generadora de autómatas estocásticos de N estados
def _gen_probVect(N=16):
    vect=np.random.sample(N)
    _normVect(vect)
    return vect

# Suma pesada de dos vectores según la transformación de Lorentz de velocidades
def _LorentzSum(M,P,lambd):
    if np.shape(np.nonzero(M)[0])[0] != np.shape(M)[0]:
        return M # Si M está en la frontera no se efectúa ningún cálculo
    if np.all(M == P):
        return M # Si M y P son iguales no hay nada que hacer
    delta=M-P
    # A partir de la recta paramétrica que pasa por M con vector director PM
    # se determinan los puntos Alfa y Omega, descritos por los valores tneg y
    # tpos respectivamente. Con estas consideraciones la función kappa se simplifica.
    nzero=np.nonzero(delta)[0]
```

```python
    t=-M[nzero]/delta[nzero]
    tneg=-np.max(t[t<0])
    tpos=np.min(t[t>0])
    output=M+lambd*delta*( (2*tneg-1)/(1+tneg*(tneg-1)/(tneg+tpos))-tneg )
    output[0]=1-np.sum(output)+output[0] # Evita errores de coma flotante
    if np.any(output < 0):
        print(t)
        print(M)
        print(P)
        print(delta)
        np.gasdgadgas
    return output

# Función análoga a __LorentzSum en una dimensión
def _c(m,p,lambd):
    output=m
    output[m>p]=(m+lambd*( (2*m-p)/(1+m*m-m*p)-m ))[m>p]
    output[m<p]=(m+lambd*( 1-(1-2*m+p)/(1+p-m+m*m-m*p)-m ))[m<p]
    return output

# Aplicación de __LorentzSum a las filas de dos matrices dadas
def _MLorentzSum(M,P,lambd):
    output=M
    for i in range(np.shape(M)[0]):
        output[i,:]=_LorentzSum(M[i,:], P[i,:], lambd)
    return output


# AUTÓMATAS DE MOORE FINITOS (ESTOCÁSTICOS)
# Autómatas implementados tal y como se describe en la memoria del TFG.
class Moore(Strategy):
    def __init__(self,Mooreinput=16):
        if type(Mooreinput)==int: # Se genera un autómata de Mooreinput estados
            self.score = 0
            self.state = -1
            self.actions = np.random.sample(Mooreinput)
```

```
            self.state0 = _gen_probVect(Mooreinput)
            matrix=np.random.sample((Mooreinput,Mooreinput))
            for i in range(matrix.shape[0]):
                _normVect(matrix[i,:])
            self.trans0 = matrix
            matrix=np.random.sample((Mooreinput,Mooreinput))
            for i in range(matrix.shape[0]):
                _normVect(matrix[i,:])
            self.trans1 = matrix
        else: # El autómata está definido manualmente
            self.score = Mooreinput['score']
            self.state = Mooreinput['state']
            self.actions = Mooreinput['actions']
            self.state0 = Mooreinput['state0']
            self.trans0 = Mooreinput['trans0']
            self.trans1 = Mooreinput['trans1']
        self.n_of_states = int(self.actions.shape[0])


    # Método que devuelve la acción llevada a cabo por el autómata
    def action(self):
        if self.state == -1:
            self.state = random.choices(range(self.n_of_states), self.state0)[0]
        p1=self.actions[self.state]
        return random.choices(range(2), [1-p1,p1])[0]

    # Método que provoca la transición del autómata
    def update(self,myaction,action2):
        aux=self.state
        if action2 == 0:
            self.state = random.choices(range(self.n_of_states), self.trans0[int(aux),:])[0]
        if action2 == 1:
            self.state = random.choices(range(self.n_of_states), self.trans1[int(aux),:])[0]

    # Método que devuelve el autómata a la configuración previa a la inicialización
```

```python
    def reset(self):
        self.state=-1


    # Método que introduce variaciones aleatorias en el código del autómata
    def _mutate(self):
        # actions
        memo=self.actions
        l=np.shape(self.actions)
        binarray=np.random.choice(2, l, p=[0.995,0.005])
        self.actions = (1-binarray)*memo + binarray*np.random.sample(l)
        _normVect(self.actions)
        # state0
        memo=self.state0
        l=np.shape(self.state0)
        binarray=np.random.choice(2, l, p=[0.995,0.005])
        self.state0 = (1-binarray)*memo + binarray*np.random.sample(l)
        _normVect(self.state0)
        # trans0
        memo=self.trans0
        l=np.shape(self.trans0)
        binarray=np.random.choice(2, l, p=[0.995,0.005])
        self.trans0 = (1-binarray)*memo + binarray*np.random.sample(l)
        for i in range(self.trans0.shape[0]):
            _normVect(self.trans0[i,:])
        # trans1
        memo=self.trans1
        l=np.shape(self.trans1)
        binarray=np.random.choice(2, l, p=[0.995,0.005])
        self.trans1 = (1-binarray)*memo + binarray*np.random.sample(l)
        for i in range(self.trans1.shape[0]):
            _normVect(self.trans1[i,:])
```

```
    # Método que devuelve dos autómatas obtenidos a partir del código de los
    # dos introducidos como argumento.
    def _crossover(self,parent1,parent2,w):
        son=Moore({'score':0,'state':-1,
        'actions':_c((parent1.actions+parent2.actions)/2, self.actions, w),
        'state0':_LorentzSum((parent1.state0+parent2.state0)/2, self.state0, w),
        'trans0':_MLorentzSum((parent1.trans0+parent2.trans0)/2, self.trans0, w),
        'trans1':_MLorentzSum((parent1.trans1+parent2.trans1)/2, self.trans1, w)})
        return son



# Función normalizadora de los filas (a partir del segundo término) de matrices
def _normMoore(matrix):
    for i in range(matrix.shape[0]):
        norm=np.sum(matrix[i,1:])
        if norm==0 or norm==1:
            continue
        matrix[i,1:]/=norm
    return matrix

# Función generadora de autómatas estocásticos de N estados
def _gen_Stoch(N=16):
    matrix=np.random.sample((N+1,N+1))
    matrix[0,0]=0
    _normMoore(matrix)
    return matrix


# AUTÓMATAS ESTOCÁSTICOS
# Son matrices cuadradas de orden n+1 donde el primer término de la primera
# fila representa el estado actual del autómata; el resto de los i-ésimos
```

```
# términos de la primera fila representan las probabilidades de que el estado i
# sea el estado inicial; cada una de las demás filas representa un estado,
# donde el primer término es la acción a llevar a cabo y los demás representan
# las probabilidades de transición a uno de los n estados del autómata.
# Estos autómatas son similares a los anteriores con la peculiaridad de que
# no se ven influidos por las acciones llevadas a cabo por otros jugadores.
# Representan estrategias no condicionadas.
class Stoch(Strategy):
    def __init__(self,Stochinput=16):
        if type(Stochinput)==int: # Se genera un autómata de Stochinput estados
            dictionary={'score':0,'code':_gen_Stoch(Stochinput)}
        else: # El autómata está definido manualmente
            dictionary={'score':0,'code':Stochinput}
        self.score = dictionary['score']
        self.code = dictionary['code']
        self.n_of_states = self.code.shape[0]-1


    # Método que devuelve la acción llevada a cabo por el autómata
    def action(self):
        if self.code[0,0] == 0:
            self.code[0,0] = 1+random.choices(range(self.n_of_states), self.code[0,1:])[0]
        p1=self.code[int(self.code[0,0]) , 0]
        return random.choices(range(2), [1-p1,p1])[0]

    # Método que provoca la transición del autómata
    def update(self,myaction,action2):
        aux=self.code[0,0]
        self.code[0,0] = 1+random.choices(range(self.n_of_states), self.code[int(aux),1:])[0]
```

```

    # Método que devuelve el autómata a la configuración previa a
    ↪ la inicialización
    def reset(self):
        self.code[0,0]=0


    # Método que introduce variaciones aleatorias en el código
    ↪ del autómata
    def _mutate(self):
        memo=self.code
        binmatrix=np.random.choice(2, self.code.shape,
        ↪ p=[0.995,0.005])
        self.code = (1-binmatrix)*memo +
        ↪ binmatrix*np.random.sample(self.code.shape)
        self.code[0,0]=memo[0,0]
        self.code=_normMoore(self.code)

    # Método que devuelve dos autómatas obtenidos a partir del de
    ↪ los dos
    # introducidos como argumento.
    def _crossover(self,parent1,parent2,w):
        soninput=self.code
        soninput[:,1:]=
        ↪ _MLorentzSum((parent1.code[:,1:]+parent2.code[:,1:])/2,
        ↪ self.code[:,1:], w)
        soninput[1:,0]=
        ↪ _c((parent1.code[1:,0]+parent2.code[1:,0])/2,
        ↪ self.code[1:,0], w)
        son=Stoch(soninput)
        return son



# AUTÓMATAS DE MOORE FINITOS (BINARIOS)
# Están implementados siguiendo Miller, 1993. Consisten en una
↪ cadena de hasta 148
# bits en la que los 4 primeros bits definen el estado inicial
↪ del autómata.
```

```python
# Las siguientes secuencias de 9 bits se corresponden con: el primer bit define
# la acción tomada por el autómata; los siguientes 4 bits determinan a qué
# estado transiciona el autómata en caso de cooperación por parte del rival;
# finalmente los últimos 4, el estado en caso de no cooperación. De esta forma
# se representan automatas de Moore finitos.
class Miller(Strategy):
    def __init__(self,Millerinput=16):
        if type(Millerinput)==int: # Se genera un autómata de Millerinput estados
            dictionary= {'score':0, 'code':np.random.randint(2,size=Millerinput*9+4)}
        else: # El autómata está definido manualmente
            dictionary={'score':0,'code':Millerinput}
        self.score = dictionary['score']
        self.code = dictionary['code']
        self.n_of_states = (self.code.shape[0]-4)/9
        self.code_len = self.code.shape[0]
        self._state = -1


    # Método que devuelve la acción llevada a cabo por el autómata
    def action(self):
        if self._state == -1:
            self._state = 4+9*np.inner([8,4,2,1], self.code[0:4])
        return self.code[self._state]

    # Método que provoca la transición del autómata
    def update(self,myaction,action2):
        if action2 == 0:
            self._state = 4+9*np.inner([8,4,2,1], self.code[self._state+1:self._state+5])
        if action2 == 1:
            self._state = 4+9*np.inner([8,4,2,1], self.code[self._state+5:self._state+9])

```

```python
    # Método que devuelve el autómata a la configuración previa a la inicialización
    def reset(self):
        self._state=-1


    # Método que introduce variaciones aleatorias en el código del autómata
    def _mutate(self):
        self.code = ( self.code + random.choices(range(2), k=self.code_len, weights=[0.995,0.005]) )%2

    # Método que devuelve dos autómatas obtenidos a partir del código de los
    # dos introducidos como argumento.
    def _crossover(self,parent1,parent2,w):
        son=copy.deepcopy(parent1)
        c=np.random.randint(0,self.code_len) # Pivote
        l=np.random.randint(1,self.code_len) # Longitud del intercambio
        iterator=itertools.cycle(range(self.code_len))
        control=0
        for i in range(c): # Los intercambios comienzan en el bit c
            next(iterator)
        for i in iterator: # Proceso de intercambio
            son.code[i]=parent2.code[i]
            control+=1
            if control==l:
                break
        return son



# Función que devuelve los índices de los n elementos más pequeños de una lista l
def _mins(l, n):
    return list(np.where(l <= np.sort(l)[n-1])[0])[:n]


```

```
# Una población viene dada por un vector numpy cuyas coordenadas son células
class Population():
    def __init__(self,data,n=16,N=30):
        if type(data)==type: # Se genera una población de N células de clase data
            self.cells = np.array([data(n) for i in range(N)])
        else: # La población está definido manualmente
            self.cells = data

    def evolve(self,test,steps=50,renewprop=1/3): #El máximo de renewprop es de hecho 1/3
        limit=self.cells.shape[0]
        extint=int(np.floor(limit*renewprop))
        for k in range(steps):
            mu=np.zeros(limit,dtype=int) # Vector de pagos
            # Battle Royale y Mirror Match
            indexes = np.column_stack(np.triu_indices(limit, k=0))
            indexes1=indexes[:,0]
            indexes2=indexes[:,1]
            cells_1=np.array(self.cells[indexes1])
            cells_2=np.array(self.cells[indexes2])
            #aux1,aux2=zip(*[test(player1, player2).score for player1, player2 in zip(cells_1, cells_2)])
            resultados = np.array([[elemento.score for elemento in test(player1,player2)] for player1, player2 in zip(cells_1, cells_2)])
            for j in range(limit):
                mu[j]+=np.sum(resultados[:,0][indexes1==j])
                mu[j]+=np.sum(resultados[:,1][indexes2==j])
            # Normalizacion de los pesos probabilisticos
            dev=np.std(mu)
            if dev == 0:
                print(k+1,np.mean(mu)) # Muestra la puntuación media
                break
            m=np.mean(mu)
            # Pesos para el crossover
            d=(m-mu)/dev
            d[d>1]=1
```

```
            d[d<0]=d[d<0]/3
            d[d<-1]=-1/3
            w=(mu-m)/dev+2 # Pesos normalizados
            w[w < 0]=0. # No negatividad
            windex=_mins(w,extint) # Indices de los w mas pequeños
            for i in windex:
                w[i]=0
            total=np.sum(w)
            w/=total
            # Iteracion evolutiva
            for i in windex:
                p=np.random.choice(limit, 2, replace=False, p=w) # Celulas padre
                self.cells[i]= self.cells[i]._crossover( self.cells[p[0]], self.cells[p[1]], d[i])
                self.cells[i]._mutate()
            print(k+1,m)

# Dilema del prisionero repetido infinitamente
def RPD(player1,player2,rounds=150,payoffs={ '00':[3,3],'01':[0,5],'10':[5,0],'11':[1,1]}):
    p1=copy.deepcopy(player1)
    p2=copy.deepcopy(player2)
    p1.score, p2.score = 0, 0
    for i in range(rounds):
        a1= p1.action() # Estados
        a2= p2.action()
        # Juego de etapa (PD)
        p1.score+=payoffs['{}{}'.format(a1,a2)][0]
        p2.score+=payoffs['{}{}'.format(a1,a2)][1]
        # Transiciones de estados
        p1.update(a1,a2)
        p2.update(a2,a1)
    p1.reset()
    p2.reset()
    return p1,p2
```

```
# SCRIPT
# Guión que ejecuta el algoritmo genético una vez para cada tipo
↪  de autómata
# implementado e imprime por pantalla el tiempo de cálculo.

#%% Cadena binaria
pop0=Population(Miller) # Población de células
start = time.time()
pop0.evolve(RPD) # Algoritmo genetico evolutivo para el RPD
end = time.time()
print('Time(s):', end - start)

#%% Autómata estocástico
pop1=Population(Stoch) # Población de células
start = time.time()
pop1.evolve(RPD) # Algoritmo genetico evolutivo para el RPD
end = time.time()
print('Time(s):', end - start)

#%% Autómata de Moore generalizado
pop2=Population(Moore) # Población de células
start = time.time()
pop2.evolve(RPD) # Algoritmo genetico evolutivo para el RPD
end = time.time()
print('Time(s):', end - start)
```

Bibliografía

[1] Alonso-Sanz R., *A quantum prisoner's dilemma cellular automaton*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 470.2164 (2014): 20130793.

[2] Bertelle C., et al, *Adaptive behaviour for prisoner dilemma strategies based on automata with multiplicities*, ESS 2002, 14th European Simulation Symposium and Exhibition, Oct 2002, Dresden, Germany.

[3] Cerdá E., Jimeno J. L., Pérez J., *Teoría de juegos*, Ibergarceta Publicaciones, 2013.

[4] Friedman J.W., *A non-cooperative equilibrium for supergames*, The Review of Economic Studies 38.1 (1971): 1-12.

[5] Fundenberg D., Tirole J., *Game theory*, MIT press, 1991.

[6] García J., van Veelen M., *In and out of equilibrium I: Evolution of strategies in repeated games with discounting*, Journal of Economic Theory 161 (2016): 161-189.

[7] García J., van Veelen M., *No strategy can win in the repeated prisoner's dilemma: linking game theory and computer simulations*, Frontiers in Robotics and AI 5 (2018): 102.

[8] Kehagias Ath., *Probabilistic learning Automata and the prisoner's dilemma*, Journal of Liberal Arts 1 (1994): 41-55.

[9] Kramer O., *Genetic algorithms*, Springer International Publishing, 2017.

[10] Lui J.C.S., *Introduction to Game Theory Evolution Games Theory: Replicator Dynamics*, Notes of the Department of Computer Science and Engineering of the Chinese University of Hong Kong: http://www.cse.cuhk.edu.hk/ cslui/CSC6480/replicator.pdf

[11] Mealy George H., *A method for synthesizing sequential circuits*, The Bell System Technical Journal, 34(5), 1045-1079.

[12] Miller J.H., *The coevolution of automata in the repeated prisoner's dilemma*, Journal of Economic Behavior and Organization 29.1 (1996): 87-112.

[13] Moore E. F., *Gedanken-experiments on Sequential Machines*, Automata studies, 34, 129-153.

[14] Paz A., *Introduction to probabilistic automata*, Academic Press.

[15] Poundstone W., *Prisoner's dilemma: John von Neumann, game theory, and the puzzle of the bomb*, Anchor, 1993.

[16] Resnick R., *Introduction to Special Relativity*, 1st ed. Wiley, 1968. ISBN: 9780471717256.

[17] Schuster P., Sigmund K., *Replicator dynamics*, Journal of theoretical biology 100.3 (1983): 533-538.

[18] Sérgio A.R., Schimit P.H.T., *Interaction characteristics as evolutionary features for the spatial Prisoner's Dilemma in a population modeled by continuous probabilistic cellular automata and evolutionary algorithm*, Ecological complexity 42 (2020): 100829.

[19] Smith J.M., *Game Theory and the Evolution of Fighting*, On Evolution, Edinburgh University Press, Edinburgh, 1972.

[20] Smith J.M., Price G.R., *The logic of animal conflict*, Nature 246.5427 (1973): 15-18.

[21] Smith J.M., *The theory of games and the evolution of animal conflicts*, Journal of theoretical biology 47.1 (1974): 209-221.

[22] Taylor P.D., Jonker L.B., *Evolutionary stable strategies and game dynamics*, Mathematical biosciences 40.1-2 (1978): 145-156.

[23] Taylor P.D., *Evolutionarily stable strategies with two types of player*, Journal of applied probability 16.1 (1979): 76-83.

[24] Tomašev N., et al, *Assessing game balance with AlphaZero: Exploring alternative rule sets in chess*, arXiv preprint arXiv:2009.04374 (2020).

[25] Yoo, Y., *Kakutani's fixed point theorem and the minimax theorem in game theory*, The University of Chicago, 2016.