GENERACION DE AUTOMATAS DE TAMAÑO 3 PARA ESTRATEGIAS DOMINADAS EN JUEGOS REPETIDOS

Mario A. Silvestri * y Luis G. Quintas **

Dirección: Instituto de Matemática Aplicada San Luis - Facultad de Ciencias Físicas, Matemáticas y Naturales - Universidad Nacional de San Luis - Ejército de los Andes 950 - (5700) San Luis - Argentina.
Fax: 0652-30224
Telex: 58125 UNSL-AR
Correo Electrónico: "quintas@imasl.edu.ar"
Teléfono: 0652-22803
* Cargo: JTP, Dpto. de Informática, UNSL.
** Cargo: Profesor Titular, Dpto. de Matemáticas, UNSL.
Tópicos: Teoría de Juegos - Autómatas - Optimización y Simulación.
Area: Optimización y Simulación.
Palabras Claves: Juegos Repetidos - Autómatas - Equilibrios - Estrategias Dominadas.

RESUMEN:

El objetivo central persigue poder delegar confiablemente en programas de computación la toma de decisiones complicadas, o al menos conseguir de tales programas un asesoramiento confiable.

En este trabajo se obtiene una generación completa de autómatas de tamaño 3 que implementan estrategias en juegos repetidos simetricos, complementando el estudio de Silvestri-Quintas (1992) donde sólo se consideraba una subclase (CONEXION OBLIGADA) y generalizando los estudios hechos para autómatas de tamaño 2 por Silvestri-Quintas (1994/95).

Se consideran conductas óptimas (Equilibrios de Nash). Se estudian procesos de eliminación de estrategias dominadas.

INTRODUCCION

En el presente trabajo se generalizan, para autómatas de tamaño 3, los métodos de generación de autómatas presentados para el caso autómatas de 2 estados por Silvestri-Quintas (1994/95),

Al contar con una generación completa se pueden considerar eliminaciones de estrategias dominadas según los siguientes criterios de dominación: Fuerte, regular y débil.

Una segunda posibilidad es estudiar el Conjunto de pagos en equilibrio de Nash resultante en un juego repetido simétrico.

Daremos una breve descripción de los conceptos básicos de Teoría de Juegos que se usan en este trabajo.

Un juego modela matemáticamente una situación de conflicto estratégico donde interactúan individuos (jugadores) que tienen control parcial sobre los resultados de esta interacción por el uso de determinadas estrategias. Los resultados del juego se miden por una función de utilidad que representa la ganancia (usualmente dinero) que obtuvo cada jugador.

Un ejemplo típico lo representa el juego del Dilema de los Prisioneros (la motivación del nombre y mas detalles se pueden encontrar en Friedman (1990), Silvestri-Quintas (1994), etc).

Jugador 2

Jugador 1		C	D
	C	2, 2	-1,3
	D	3, -1	0,0

Figura 1

Aquí hay dos jugadores, cada uno tiene dos estrategias C y D, uno juega por fila cuyo pago es el primer número antes de la coma y otro que elige la columna y cuyo pago es el segundo número. Esta elección se hace simultáneamente e independientemente. Se trata de un juego simétrico ya que la matriz de los primeros pagos correspondiente al primer jugador es igual a la transpuesta de la matriz de pagos del segundo jugador.

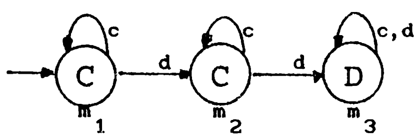
Un juego repetido consiste de la repetición infinita de un juego como el presentado anteriormente. En tal juego en cada etapa se eligen las estrategias a usar y los pagos se evalúan por medio del límite del promedio de los pagos de cada etapa.

Esto da un marco dinámico que permite el uso de estrategias globales donde los jugadores pueden reaccionar a las estrategias que en cada etapa usa su oponente. Estas estrategias globales se pueden implementar por autómatas, los cuales consisten de estados donde cada jugador actúa en una forma determinada (elige una estrategia particular para la etapa del juego que correponda jugar) y transiciones entre estados que permiten el cambio de estados del autómata de acuerdo a lo que el oponente ha jugado en la etapa anterior.

GENERACION DE AUTOMATAS DE LA SUBCLASE CONEXION OBLIGADA

La idea de este método es obligar a **linealizar** los autómatas.Esto significa que la estructura de los autómatas se verá restringida a una sola manera de conectarse, donde el estado m_1 siempre será el "estado inicial", y si existe conexión con otro estado será con el m_2, y si de éste con otro será con el m_3, y así sucesivamente. La conexión se realiza por alguna de las salidas de cada estado, esto no descarta la posibilidad de que otra salida se conecte con otro estado o consigo mismo.

La idea de linearizar se puede ver con un ejemplo de un autómata (estrategia) conectado en forma obligada, en el Dilema de los Prisioneros.



	c	d	
1	1	2	(C)
2	2	3	(C)
3	3	3	(D)

Figura 2

Vemos que el estado m_1 que contiene la acción "C" se conecta con el estado m_2 por "d", y éste con m_3 por "d". Al forzar esta unión nos aseguramos de tratar una subclase de los autómatas con estados accesibles y no tratar ningún autómata con estados inaccesibles. Así generaremos los autómatas distintos estructuralmente con estados accesibles, directamente, sin comparaciones, ni eliminaciones posteriores.

GENERACION DE AUTOMATAS DISTINTOS EN FORMA ESTRUCTURAL

La siguiente es la secuencia para generar los autómatas distintos en forma estructural de tamaño <= N de la subclase :

A) Generamos los autómatas de tamaño N, **evitando** aquellos que tengan estados inaccesibles.(es decir, no generamos y luego eliminamos comparando,sino generamos directamente los válidos)

B) Decrementamos el N en uno.

C) Si N > 0 entonces volvemos al paso A), sino fin del proceso.

Definimos una "casilla prototipo" como una matriz de transiciones donde cada posición (i,j) tendrá todos los índices de los estados con los cuales puede conectarse el estado "i" (separados con comas). Cada casilla prototipo identificará un grupo de matrices de transición de autómatas de esta subclase.

Veamos un ejemplo para N=3, recordando que sólo analizamos los distintos en forma estructural.

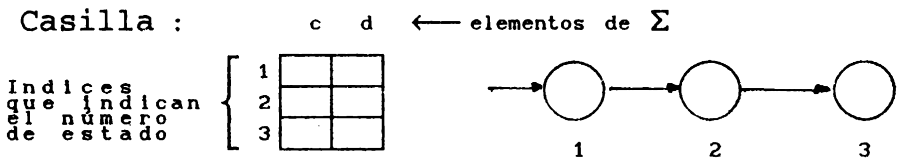
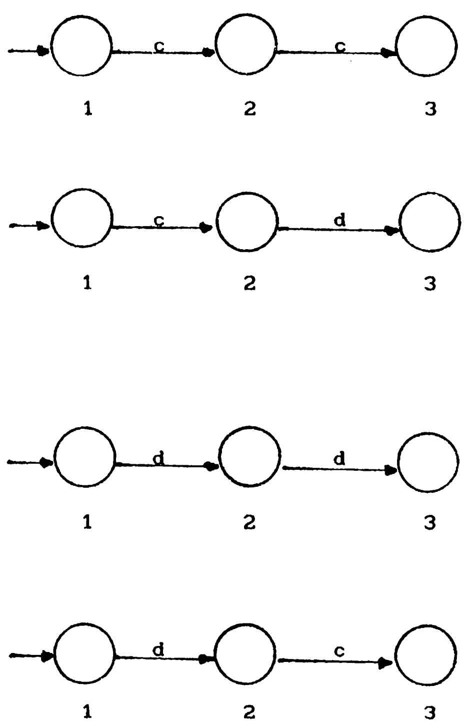


Figura 3

Obligamos una secuencia, de manera que podamos controlar la generación de casillas prototipos de matrices de transiciones que servirán para la generación de los autómatas.

Así se debe conectar el estado 1 con el estado 2 y el estado 2 con el estado 3. Veamos cuales son las distintas casillas prototipos si $|\Sigma|=2$ y $\Sigma=\{ c, d \}$



	c	d
1	2	*
2	3	*
3	*	*

	c	d
1	2	*
2	*	3
3	*	*

	c	d
1	*	2
2	*	3
3	*	*

	c	d
1	*	2
2	3	*
3	*	*

Figura 4

Tenemos cuatro tipos distintos de conexión para los autómatas de tamaño 3 (c,c; c,d; d,d; d,c). El asterisco "*" significa que la salida de los estados puede conectarse con cualquier otro, incluso consigo mismo (*=1,2,3). Vemos entonces que para N=3 tenemos 4 casillas prototipo.

Veamos para el caso de N=2 por extensión donde se puede verificar lo predicho. Entonces para N=2 tendremos 2 casillas prototipo:

	c	d
1	2	1,2
2	1,2	1,2

	c	d
1	1,2	2
2	1,2	1,2

Figura 5

En la Figura 7 se puede ver que realizando variaciones posibles de los elementos de cada posición de cada casilla, obtenemos

para la casilla de la izquierda tenemos:

(2) (1,2) (1,2) (1,2)
1 * 2 * 2 * 2 = 8

y para la casilla de la derecha tenemos:

(1,2) (2) (1,2) (1,2)
2 * 1 * 2 * 2 = 8

Es decir que en total tendremos 8+8 =16 autómatas distintos.

Si generamos lo que nos está representando cada casilla prototipo obtendremos las siguientes matrices de transición según este conectada por "C" o "D":

Conectada por "C"

	c	d
1	2	1
2	1	1

	c	d
1	2	1
2	1	2

	c	d
1	2	1
2	2	1

	c	d
1	2	1
2	2	2

	c	d
1	2	2
2	1	1

	c	d
1	2	2
2	1	2

	c	d
1	2	2
2	2	1

	c	d
1	2	2
2	2	2

Conectada por "D"

	c	d
1	1	2
2	1	1

	c	d
1	1	2
2	1	2

	c	d
1	1	2
2	2	1

	c	d
1	1	2
2	2	2

	c	d
1	2	2
2	1	1

	c	d
1	2	2
2	1	2

	c	d
1	2	2
2	2	1

	c	d
1	2	2
2	2	2

Figura 6

Vemos que los cuatro autómatas últimos generados por cada una de las casillas (la que conecta por "C" y la que conecta por "D") son equivalentes, por lo tanto estamos generando 16 autómatas donde 4 de

Conectada por "C" Conectada por "D"

Figura 7

ellos son equivalentes. Lo que haremos es considerar una sola vez el caso cuando se conecta el estado 1 con el estado 2 por los dos elementos del alfabeto al mismo tiempo, tomando este caso como una casilla especial y dejando las otras dos casillas sin la posibilidad de conectarse a la vez con el segundo estado.

Si realizamos este cambio para el caso de N=2 nos quedarán tres casillas prototipo para la generación de los autómatas de tamaño dos y son los siguientes:

	c	d
1	2	1
2	1,2	1,2

	c	d
1	1	2
2	1,2	1,2

	c	d
1	2	2
2	1,2	1,2

Figura 8

Calculando las variaciones posibles de cada casilla tendremos:

para la casilla de la izquierda:

(2)		(1)		(1,2)		(1,2)
1	*	1	*	2	*	2 = 4

para la casilla del medio:

(1)		(2)		(1,2)		(1,2)
1	*	1	*	2	*	2 = 4

para la casilla de la derecha:

(2)		(2)		(1,2)		(1,2)
1	*	1	*	2	*	2 = 4

Todas las combinaciones de las casillas dan un total de 4+4+4=12 autómatas distintos en forma estructural.

Veamos para el caso de N=3 para aclarar un poco más la idea y considerar una última condición.

	c	d
1	2	1,3
2	3	1,2,3
3	1,2,3	1,2,3

	c	d
1	1,3	2
2	3	1,2,3
3	1,2,3	1,2,3

	c	d
1	2	2
2	3	1,2,3
3	1,2,3	1,2,3

	c	d
1	2	1,3
2	1,2,3	3
3	1,2,3	1,2,3

	c	d
1	1,3	2
2	1,2,3	3
3	1,2,3	1,2,3

	c	d
1	2	2
2	1,2,3	3
3	1,2,3	1,2,3

Figura 9

Si calculamos las variaciones de cada casilla y las sumamos nos da:

$$3^3 * 2 + 3^3 * 2 + 3^3 + 3^3 * 2 + 3^3 * 2 + 3^3 = 270$$

es decir, tendremos 270 autómatas distintos estructuralmente.

Cuando no considerábamos los posibles repetidos que podían existir teníamos 4 casillas, ahora tenemos 6 porque eliminamos los repetidos que producirían los de la primera fila de la casilla. Conectamos el primer estado con el segundo de una forma forzada evitando repetición ya que tenemos una casilla especial para ello.

En este caso hemos generado 270 autómatas, supuestamente, distintos en forma estructural. Pero no hemos realizado el control de repetición para la segunda fila y si miramos las casillas generadas comprobaremos que estamos generando autómatas iguales, ya que sólo controlamos por la primer fila.

De esta manera el caso de N=3 nos quedaría así:

A)

	c	d
1	2	1,3
2	3	1,2
3	1,2,3	1,2,3

B)

	c	d
1	1,3	2
2	3	1,2
3	1,2,3	1,2,3

C)

	c	d
1	2	2
2	3	1,2
3	1,2,3	1,2,3

D)

	c	d
1	2	1,3
2	1,2	3
3	1,2,3	1,2,3

E)

	c	d
1	1,3	2
2	1,2	3
3	1,2,3	1,2,3

F)

	c	d
1	2	2
2	1,2	3
3	1,2,3	1,2,3

G)

	c	d
1	2	1,3
2	3	3
3	1,2,3	1,2,3

H)

	c	d
1	1,3	2
2	3	3
3	1,2,3	1,2,3

I)

	c	d
1	2	2
2	3	3
3	1,2,3	1,2,3

Figura 10

Si calculamos las variaciones de cada casilla y las sumamos nos da:

$$2 * (2^2 * 3^2 + 2^2 * 3^2 + 2^1 * 3^2) + 2^1 * 3^2 + 2^1 * 3^2 + 3^2 = 225$$

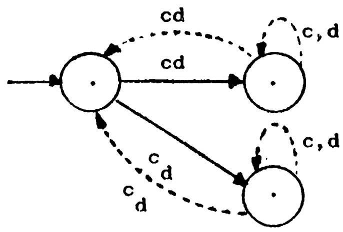
esto nos aclara que en el caso anterior estábamos considerando 45 autómatas iguales como si fueran distintos, aunque tengamos que considerar nueve casillas prototipo de generación.

Podemos ver que la casilla **C)** es el caso especial de las casillas **A)** y **B)**; la casilla **F)** es el caso especial de las **D)** y **E)** por el control de repetición de la primera fila. La casilla **G)** es el caso especial de **A)** y **D)**; la casilla **H)** es el caso especial de **B)** y **E)** por el control de repetición la segunda fila. Y por último la casilla **I)** es el caso especial de **C)**, **F)**, **G)** y **H)**.

Este proceso es sólo para obtener todas las formas estructurales distintas de la subclase, luego tenemos que agregar todas las variaciones posibles de los valores de las funciones de comportamiento de los estados, evitando aquellas en que la acción es la misma en todos los estados, por lo cual la fórmula nos quedaría $P^N - P$.

GENERACION DE AUTOMATAS DE TAMAÑO 3 - NO CONSIDERADOS EN LA SUBCLASE

Aquí consideraremos la generación de los autómatas no considerados en la subclase CONEXION OBLIGADA. El prototipo de tales autómatas tiene la siguiente forma estructural:(Figura 11)



.....indica opcional

Figura 11

Existen algunos autómatas en los cuales se conectan los estados 2 y 3, y esos ya están considerados en el análisis previo de los autómatas de tamaño menor o igual a 3. Veamos algunos ejemplos de equivalencias donde se cumple lo predicho(Figura 12 y 13).

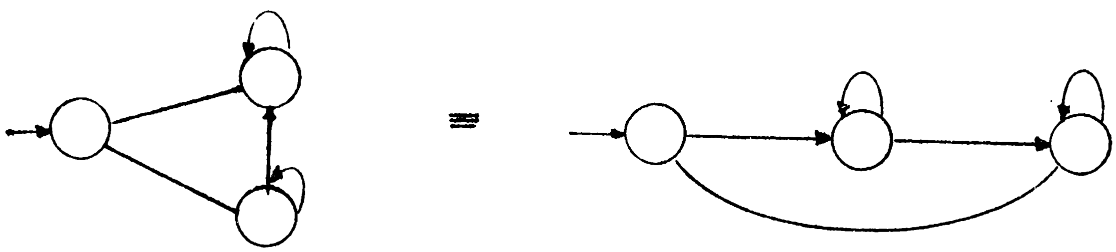
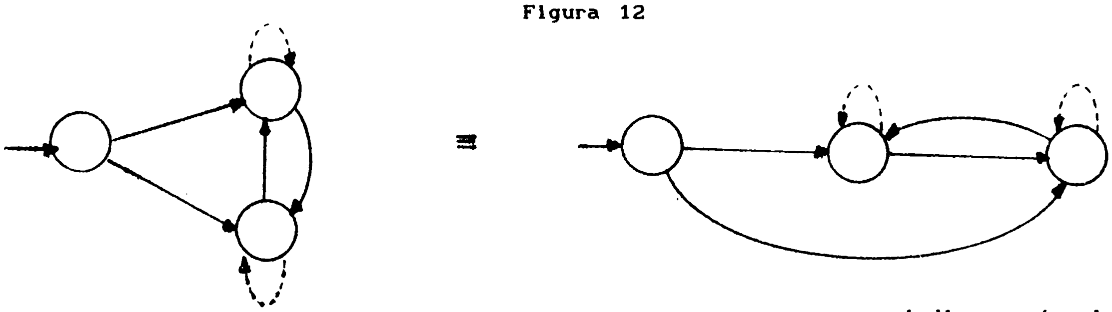


Figura 12



......indica opcional

Figura 13

Esto nos dice que solamente faltaban incluir aquellas formas estructurales donde no se conectan los estados 2 y 3; y si existe conexión entre los estados 1 con el 2 y 1 con el 3.

Entonces se deberían generar todas las formas estructurales que tienen la característica de la Figura 11 y luego, combinar las funciones comportamiento en los estados.

Si analizamos las distintas formas estructurales vemos que si consideramos una de las ramas del autómata Fig. 11, podemos realizar 12 variaciones diferentes de la forma estructural con alfabeto de salida de tamaño 2 ($\Sigma=\{c,d\}$).(Fig.14)

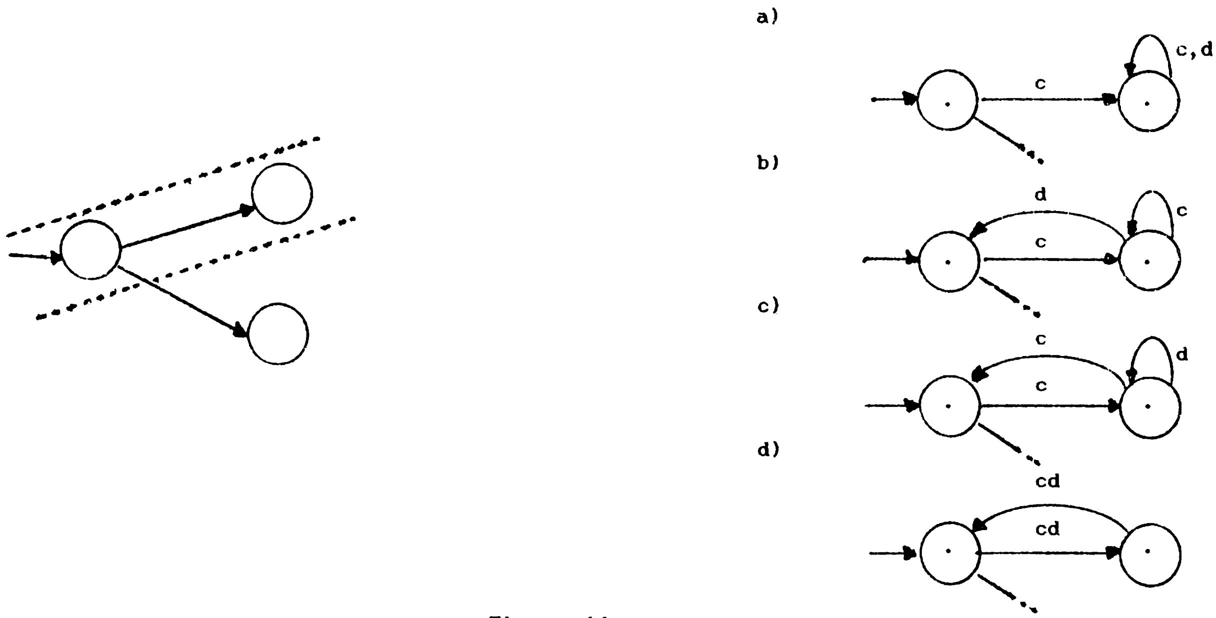


Figura 14

Pero como vamos a rotar o combinar los valores internos de cada estado por la función comportamineto, entonces no es necesario considerar el paso por "c" y luego por "d", ya que existirá una equivalencia al final de la generación:

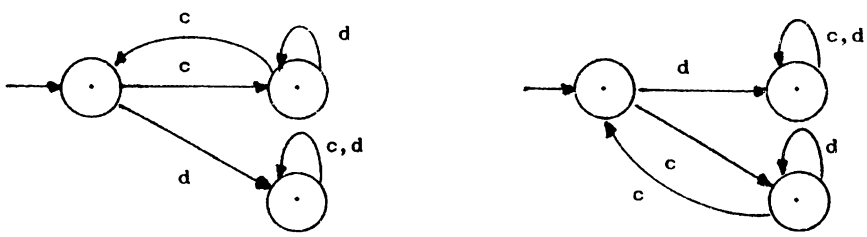


Figura 15

Así sólo tendremos que combinar 4(por c) * 4(por d) = 16 estructuras distintas.

Ahora agreguemos las acciones en los estados, eliminando los casos, en que la misma acción se encuentre presente en todos los estados a la vez.

$$P^N - P = 2^N - 2 = 2^3 - 2 = 8 - 2 = 6$$

Así la cantidad de autómatas distintos es:

$$4 * 4 * (2^3 - 2) = 96$$

La Casilla Prototipo de los autómatas no incluídos en la Subclase es la siguiente:

	c	d
1	2	3
2	1,2	1,2
3	1,3	1,3

Figura 16

Hay que tener en cuenta que se agregan 16 formas estructurales nuevas y 96 estrategias nuevas, donde el estado 1 puede conectarse con el 3, sin existir conexión del 2 con el 3. Veamos el algoritmo.

ALGORITMO

El siguiente algoritmo produce la generación de las formas estructurales de los autómatas.

```
void genera16() /* proc. genera 16 formas estructurales extras */
{
 int raux1[6][2];int k1,k2,k3,k4;
 raux1[1][0]=2;   raux1[1][1]=3;
 for (k1=1;k1<=2;k1++) {
        for (k2=1;k2<=2;k2++)   {
                raux1[2][0]=k1;raux1[2][1]=k2;
                  for (k3=1;k3<=3;k3++)  {
                        for (k4=1;k4<=3;k4++)     {
                              if ((k3!=2)&&(k4!=2))
                                 raux1[3][0]=k3;  raux1[3][1]=k4;
                                    grabacasilla(raux1);
                                              }
                                    }
                              }
                        }
}
```

Algoritmo 1

El procedimiento "grabacasilla" almacena las filas de las matrices de transición de los distintos autómatas en forma consecutiva en una Tabla "Autom" que es de tipo char. Es decir, cada posición de la tabla es un byte (8 bits) y los autómatas en nuestro estudio tendrán un tamaño menor o igual a 3 por lo que se realiza una conversión de dos números enteros en un char (Ver Silvestri (1992) y así se reduce el espacio de almacenamiento.

Por último, con una gráfica aclaremos la idea de como quedan almacenados los distintos autómatas de tamaño menor o igual a 3 a partir de la posición cero de la Tabla "Autom". (Figura 18)

Resultan del algoritmo las siguientes formas estructurales de los autómatas: (Figura 17)

	c	d			c	d			c	d			c	d
1	2	3		1	2	3		1	2	3		1	2	3
2	1	1		2	1	1		2	1	1		2	1	1
3	1	1		3	1	3		3	3	1		3	3	3

	c	d			c	d			c	d			c	d
1	2	3		1	2	3		1	2	3		1	2	3
2	1	2		2	1	2		2	1	2		2	1	2
3	1	1		3	1	3		3	3	1		3	3	3

	c	d			c	d			c	d			c	d
1	2	3		1	2	3		1	2	3		1	2	3
2	2	1		2	2	1		2	2	1		2	2	1
3	1	1		3	1	3		3	3	1		3	3	3

	c	d			c	d			c	d			c	d
1	2	3		1	2	3		1	2	3		1	2	3
2	2	2		2	2	2		2	2	2		2	2	2
3	1	1		3	1	3		3	3	1		3	3	3

Figura 17

AUTOM:

TAMAÑO 1	0	0	1	1
	1	1	2	1
		2	1	1
	2	3	2	1
		4	1	2
	3	5	2	1
		6	2	1
TAMAÑO 2	4	7	2	1
		8	2	2
	5	9	2	2
		10	1	1
	6	11	2	2
		12	1	2
	7	13	2	2
		14	2	1
	⋮	⋮		

Figura 18

Así cuando incluyamos los de tamaño 3, cada uno de ellos ocupará 3 posiciones char contigüas. Debemos acceder a esta tabla de estructuras para identificar un autómata índice de la Matriz Global. La idea es, teniendo un índice de la Matriz Global poder determinar el tamaño, luego la forma y por último las acciones internas de los estados del autómata al cual representa. Veamos como varían para cada uno de los tamaños las acciones internas considerando que cada tamaño me define la cantidad de posiciones donde cada una puede tener dos valores posibles (binario).

N=1 $2^1 = 2$ combinaciones posibles [0,1]

N=2 $2^2 = 4$ " " [0,3]

N=3 $2^3 = 8$ " " [0,7]

y así sucesivamente, donde cada posición tendrá un 1 ó 0 (C ó D). Es decir, las variaciones de N posiciones con dos posibilidades cada una es 2^N, equivalente tener los números binarios $[0,2^N-1]$. En el caso de N=2 tendremos :

$$N=2 \qquad 2^2=4 \qquad [0,3] : \begin{bmatrix} 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$$

Pero en la forma que analizaremos, eliminaremos los casos equivalentes, cuando N>1 :

$$1111\ldots.1 \equiv 1$$
$$0000\ldots.0 \equiv 0$$

es decir, cuando el tamaño es mayor que 1 y sus acciones de comportamiento son iguales, son equivalentes al autómata de tamaño uno correspondiente. De esta manera tenemos 2^N-2 distintos. En nuestro caso de N=2 eliminaremos el "00" (0) y el "11" (3).

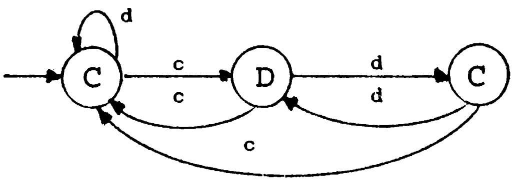
Por ejemplo, si tenemos el índice 231, significará que tenemos un cierto autómata con forma estructural y acciones en sus estados.

TAMAÑO	CANTIDAD FORMAS	CANTIDAD PERMUTACIONES	CANTIDAD DISTINTOS	CANTIDAD ACUMULADA	INDICE
1	1	$2^1 = 2$	2	2	< 231
2	12	$2^2 - 2 = 2$	24	26	< 231
3	225+16	$2^3 - 2 = 6$	1.350+96	1.471	> 231
4	5.488	$2^4 - 2 = 14$	76.832	78.304	> 231
⋮	⋮	⋮	⋮	⋮	⋮

Tabla 1: Tabla de cantidades crecientes de autómatas distintos

El primer número de la cantidad acumulada mayor que el valor índice nos está diciendo que el tamaño es el correspondiente a tal cantidad acumulada. Vemos que en la posición 26 comienza el N=3, ya que de la 0 a la 25 se encuentran los correspondientes a N=1 y N=2. Lo que hacemos es restar al 231 los 26 primeros, y nos quedan 205. A este valor lo dividimos por $2^3-2= 6$, para indicar que es el autómata número 34 de tamaño 3 y el resto de esta división al ser 1 nos dice que es la segunda combinación de acciones. Esto último es porque no consideramos el cero, entonces cuando la función módulo nos devuelve 0 (cero), nos indica que es la combinación 001 (1) y cuando nos devuelve 1 es la combinación 010 (2).

De esta manera el autómata identificado es el de la Figura 19. Cabe aclarar que en este último paso de identificación de los autómatas ya se encuentran incluídos los que no pertenecían a la Subclase, o sea que esta es la generación completa.



	c	d
1	2	1
2	1	3
3	1	2

Figura 19

EJEMPLO ILUSTRATIVO

Las siguientes figuras muestran pantallas del sistema para el Dilema de los Prisioneros con la Matriz de Pagos de la Figura 1. La Matriz Global de Pagos para autómatas de tamaño <= 3 es de 1471 filas por 1471 columnas, donde cada posición contiene 2 números reales. Esta y la secuencia de eliminación Débil en Orden Fila-Columna resultan demasiado grandes para incluirse aquí, pero se puede observar una reducción notable tanto del número de filas como el número de columnas a 1017, ya que hay 454 filas y 454 columnas eliminadas con este tipo de eliminación.

Además agregamos una muestra de la pantalla del Area de Resultados Factibles, el Area del Folk Theorem (resultados de equilibrios de Nash) y un ejemplo de autómatas en equilibrio que no pertenece a la subclase CONEXION OBLIGADA.

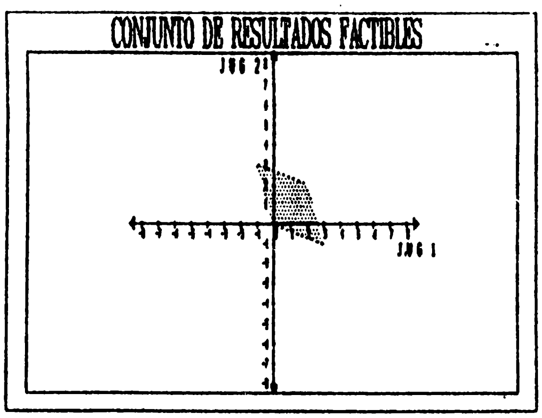
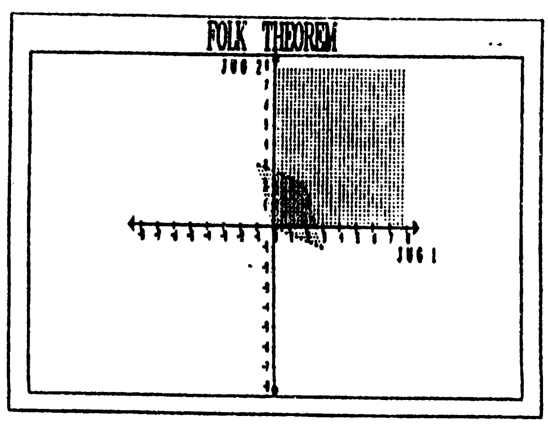


Figura 20: Pantalla del Area de Resultados Factibles.

Figura 21: Pantalla del Area del Folk Theorem.



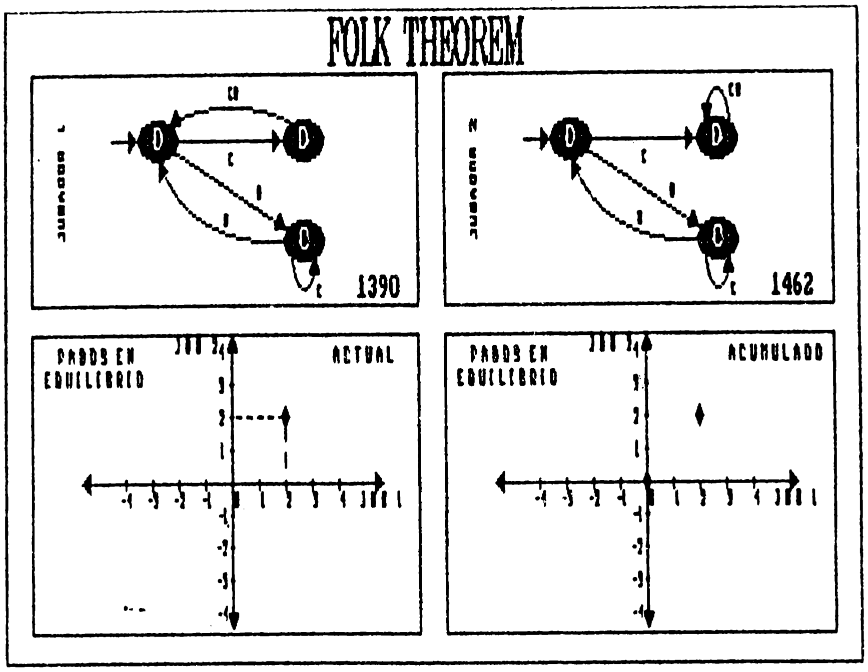


Figura 22: Ejemplo de Autómatas en Equilibrio que no pertenecen a la SUBCLASE CONEXION OBLIGADA (los últimos considerados en el análisis.

CONCLUSIONES

El desarrollo del presente sistema permitió obtener equilibrios que no fueron considerados en tratamientos anteriores.

Se pudo observar también una substancial reducción de las estrategias resultantes en el ejemplo de El Dilema de los Prisioneros cuando se consideran dominaciones Regular y Débil (no así bajo dominación Fuerte).

BIBLIOGRAFIA

Aumann R. [1981], "Survey of Repeated games". In Essay in Game Theory and Mathematical Economics in Honor of Oskar Morgenstern, Bibliographisches Institut Mannheim, Wein, Zurich, 11-42.

Friedman J. [1990], "Game Theory with Applications to Economics", Oxford Unversity Press, 2nd. edition.

Harrison [1981], "Introduction to switching and automata theory".

Hopcroft J. and Ullman J. [1979], "Introduction to Automata Theory, Languages and Computation" Reading, Massachusetts. Adison-Wesley

Kalai E. [1987], "Artificial Decisions and strategic complexity in repeated games", Proceeding of the International Workshop on Game Theory, Columbus, Ohio.

Moore E. [1956], "Experiments on Sequencial Machines" in Automata Studies, Princeton University Press.

Silvestri M. [1992], "Aplicación de autómatas en procesos de eliminación de estrategias dominadas en juegos repetidos" Tesis de Licenciatura en Ciencias de la Computación. Fac. de Cs. Físico Matemática y Naturales - U.N.S.L.

Silvestri-Quintas [1994], "Aplicación de autómatas en procesos de eliminación de estrategias dominadas en juegos repetidos" Anales 23 JAIIO, 39-53.

Silvestri-Quintas [1995], "Uso de Autómatas en procesos de Decisión" Anales INFOCOM'95, 240-252.