

Resolución Deductiva del Kakuro en CLIPS

Ingeniería del Conocimiento

Rafael Caballero González

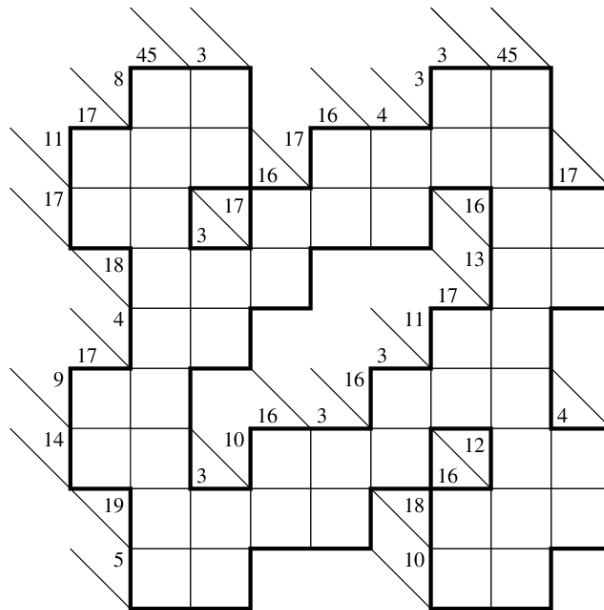
Curso 2017/2018

Contenido

1. Introducción al problema	2
2. Software usado	3
3. Estrategia y reglas implementadas	3
a. Reglas de prioridad alta	3
b. Reglas de prioridad media	4
c. Reglas de prioridad baja	6
4. Resultados	7
5. Bibliografía	7

1. Introducción al problema

El pasatiempo Kakuro es uno de los rompecabezas lógicos más extendidos del mundo junto al Sudoku. Para la asignatura Ingeniería del Conocimiento del Máster Universitario en Lógica, Computación e Inteligencia Artificial se propone como trabajo la resolución deductiva mediante CLIPS de estos pasatiempos.



El objetivo de este puzle no es otro que el de situar cifras del 1 al 9 en las casillas del tablero.

Las reglas para colocar esos números son las siguientes:

- Los números de casillas contiguas deben sumar el valor que aparece en el margen (restricción).
- En dichas casillas contiguas no pueden aparecer números duplicados.

Se dice que el Kakuro está bien planteado en caso de que tenga solución única.

Para este trabajo, se proporciona un fichero con 50 Kakuros bien planteados con tableros 9x9. Además, se proporciona un fichero base en el que se plantea la estructura de datos y se lee los Kakuros a resolver.

2. Software usado

Para la resolución de este problema se ha trabajado con el siguiente software:

- Windows 10 64bits
- Editor Atom con el plugin language-clips que asiste al desarrollo con CLIPS.
- CLIPS 64bits en su versión precompilada para Windows.
- GitKraken para el control de versiones.

3. Estrategia y reglas implementadas

En el presente apartado se explica la estrategia de resolución global del problema y se explican las reglas usadas agrupadas, puesto que muchas de ellas son similares entre sí.

La estrategia que se ha seguido ha sido en todo momento eliminar la mayor cantidad de números posibles dentro del rango de cada celda. En ningún momento se han almacenado posibles combinaciones entre celdas ni nada parecido.

Por otro lado, las reglas han sido pensadas para imitar como un humano resolvería este problema: se comienza con lo evidente y cuando la cosa se pone más complicada, se analiza lo que se tiene.

Las reglas se pueden agrupar de forma global en 3 grupos según su prioridad:

- Reglas de prioridad alta
- Reglas de prioridad media
- Reglas de prioridad baja

A continuación se explica cada uno de los grupos.

a. Reglas de prioridad alta

El conjunto de reglas implementadas en este grupo son reglas que tienen que ser ejecutadas tan pronto como sea posible. Son las encargadas de eliminar lo imposible y/o evidente.

Dentro de este grupo encontramos las reglas:

- **elimina-basura.** Es la encargada de eliminar números del rango no usados para alcanzar la suma de la restricción. Para ello hace uso del template definido “basura” que consiste en una variable simple para el id y una variable múltiple para almacenar varios valores. Esta regla buscará una celda que tenga el id de basura y eliminará todos los números no encontrados en el rango de basura.

Está conectada con las reglas de baja prioridad, que son las encargadas de proporcionar la basura, por lo que se explicará más adelante con mayor detalle cómo se completa todo el proceso.

- **elimina-mayores-a-restricciones.** Como su nombre indica, elimina los valores del rango mayores o iguales a los de las posibles restricciones.
- **elimina-asignados-fila/columna.** Estas dos reglas se activan cuando una celda es resuelta y se encargan de eliminar el número asignado del resto de celdas de la misma fila o columna.
- **elimina-mitad-si-par-menor-igual-18.** En caso de existir una restricción par e igual o menor que 18 para 2 casillas, se eliminan el valor de la restricción entre 2 de ambas casillas por ser una solución imposible. Por ejemplo: Restricción de valor 8 para 2 casillas, se elimina el 4 de ambas.

b. Reglas de prioridad media

Las reglas de prioridad media no tienen una prioridad como tal asignadas, sino que se deja a CLIPS que decida cuando aplicará cada una de ellos. Son reglas que no interfieren entre sí. Esto quiere decir que, aunque la activación de una puede provocar la activación de otra, el resultado final será el mismo. Algunas de estas reglas son aplicables a la resolución del Sudoku, aunque han sido adaptadas para el Kakuro.

Dentro de este grupo encontramos las reglas:

- **restricción-2-a-2-de-17-16.** Es una de las reglas usadas para comenzar resolver algunos Kakuro, ya que ocurren en un caso especial como el mostrado en la siguiente imagen:

	16	15	
17			4
15			
	3		

En este caso, existe una restricción de valor 16 con 2 celdas y una restricción de valor 17 con dos celdas. Ambas restricciones comparten una de sus dos celdas.

En tal caso, la celda compartida solo puede ser 9, por lo que se asigna ese valor a su rango, borrando todos los demás.

- **par/trio-asignado-fila/columna.** Estas reglas implementan la misma técnica usada en la resolución del Sudoku, en las que, en el caso de los pares, existen para una restricción dos celdas con solo dos posibles valores. En este caso, estos dos valores pueden ser eliminados del resto de celdas de la restricción. Lo mismo aplica para los tríos, solo que con tres valores y casillas.
- **max/min-valores-2/3/4/5/6.** Este conjunto de reglas intenta resolver una restricción de 2, 3, 4, 5 o 6 celdas probando con los valores máximos de cada uno o los valores mínimos. En caso de que la suma de alguno de los dos conjuntos resulte exactamente en el valor de la restricción, se pueden resolver todas las celdas de la restricción, dando por bueno el resultado.
- **detecta-cuadro-con-1-extra-1/2/3/4.** Consisten en 4 reglas encargadas de localizar la siguiente situación ilustrada en la figura:

	10	13		
17	A	B	8	3
7	C	D	E	11 17
		14		

Hay que encontrar 4 restricciones (en el caso del ejemplo serían las de valores 10, 13, 17 y 7) que tengan 2 celdas, 2 celdas, 2 celdas y 3 celdas, respectivamente. Además, deben compartir celdas como sucede con A, B, C y D.

En tal caso, se puede conocer el valor de la casilla E operando $(17+7)-(10+13) = 1$.

Se vaciaría el rango de la celda E colocando solamente el valor 1.

Existen 4 reglas de este tipo para localizar distintas posibilidades de que la celda E se encuentre en otros lugares.

- **completa-2/3/4/5/6.** Se trata de un conjunto de reglas encargadas de localizar restricciones de 2, 3, 4, 5 o 6 celdas en las que solo quede una celda por resolver. En tal caso, se resuelve dicha celda con sumando el resto de celdas resueltas y restándoselo al valor de la restricción.
- **suma-unica-2c/3c/4c/5c-{valor}.** Estas reglas se basan en que existen varios valores en las restricciones para los que solo puede existir una combinación limitada de números. Para ello, se han implementado sumas únicas de 2, 3, 4 y 5 celdas que buscan esos valores de restricción y eliminan todo número imposible del rango de las celdas. En la siguiente tabla se muestran los valores programados con los únicos valores permitidos en las celdas:

Número de celdas	Valor de la restricción	Posibles valores
2	3	1, 2
2	4	1, 3
2	16	9, 7
2	17	9, 8
3	6	1, 2, 3
3	7	1, 2, 4
3	23	9, 8, 6
3	24	9, 8, 7
4	10	1, 2, 3, 4
4	11	1, 2, 3, 5
4	29	9, 8, 7, 5
4	30	9, 8, 7, 6
5	15	1, 2, 3, 4, 5
5	16	1, 2, 3, 4, 6
5	34	9, 8, 7, 6, 4
5	35	9, 8, 7, 6, 5

c. Reglas de prioridad baja

Por último, las reglas de prioridad baja se trata de reglas que se deben aplicar cuando no se posible seguir aplicando el resto. Esto ocurre porque, como se verá a continuación, estas reglas probarán combinaciones con los valores restantes de los rangos de las celdas. Computacionalmente requieren más procesamiento y son más efectivas cuantas menos posibilidades existan dentro de esos rangos de las celdas, por supuesto.

- **busca-candidato-unico-2c/3c/4c/5c.** Estas 4 reglas buscarán restricciones de 2, 3, 4 y 5 casillas que estén sin resolver completamente y realizarán un bucle probando los valores restantes en el rango de sus celdas para ver si cumplen la restricción. Al finalizar, en caso de que solo se haya encontrado una única solución, se resuelven las celdas de dicha restricción.

Paralelamente, al realizar los bucles, se van almacenando los valores del rango de cada celda usados para resolver la restricción satisfactoriamente. Al finalizar, si se ha encontrado más de una solución posible y la longitud de dicha variable múltiple es menor que la longitud original del rango de la celda, quiere decir que existe basura, puesto que hay valores en el rango que no han sido usados para resolver la restricción satisfactoriamente de ninguna forma. En tal caso, se introduce una estructura “basura” en memoria con la id de la celda y con rango los valores que SI deben permanecer. Esto activará la regla de alta prioridad **elimina-basura** que limpiará el rango de la celda de valores inútiles.

4. Resultados

Aplicando todas las reglas descritas en el apartado anterior, los resultados obtenidos para la resolución de los 50 Kakuro propuestos son los siguientes:

Resultados									
1	✓	11	✓	21	✓	31	✓	41	✓
2	✓	12	✓	22	✓	32	✓	42	✗
3	✓	13	✓	23	✓	33	✓	43	✗
4	✓	14	✓	24	✓	34	✓	44	✗
5	✓	15	✓	25	✓	35	✓	45	✓
6	✓	16	✓	26	✓	36	✓	46	✓
7	✓	17	✓	27	✓	37	✓	47	✗
8	✓	18	✓	28	✓	38	✓	48	✓
9	✓	19	✓	29	✓	39	✓	49	✗
10	✓	20	✓	30	✓	40	✓	50	✓

Comentar que el tiempo de ejecución es prácticamente instantáneo sin importar el número del Kakuro seleccionado. Si bien este tiempo podría reducirse, por cada regla activada se imprime un mensaje por pantalla con el comando “printout” que informa de la activación de la misma así como de la casillas y/o valores a los que afecta. Estos mensajes han sido muy útiles durante el desarrollo del sistema y también lo son para ilustrar el proceso de resolución completo. Se entiende que la impresión de estos mensajes ralentiza el sistema, aunque se puede omitir.

Los Kakuro que no han conseguido resolverse completamente, terminan con pocas casillas de incógnitas. Es decir, su resolución es prácticamente completa a falta de pocas celdas.

5. Bibliografía

Para la resolución de este proyecto se ha usado de referencia lo siguiente:

- Material de clase de la asignatura Ingeniería del Conocimiento 17/18 – MULCIA – Universidad de Sevilla (transparencias y ejercicios resueltos).
- Guía de referencia para CLIPS ([link](#)).
- Estrategias de resolución del Kakuro ([link1](#)) ([link2](#)) ([link3](#)) ([link4](#)).