

REPRESENTACIÓN Y SIMPLIFICACIÓN DE SUDOKUS EN PROLOG

Práctica 1: Conocimiento y Razonamiento Automatizado

(Adaptado para la asignatura de Algorítmica y Complejidad, Curso 2024-25)

Preámbulo

Un Sudoku está formado por una cuadrícula de 9x9 casillas, dividida en nueve cuadros 3x3. A partir de algunos números (del 1 al 9) ya colocados, hay que completar las casillas vacías con dígitos del 1 al 9 de tal forma que estos no deben repetirse en una misma fila, columna o cuadro 3x3.

No discutiremos cómo se puede garantizar que exista solución única a un sudoku. Por otro lado, el objetivo de la práctica no es resolver sudokus: en el manual de SWI-Prolog (<http://www.swi-prolog.org/man/clpfd.html>) se dice cómo hacerlo usando la librería **clpfd** (*Constraint Logic Programming over Finite Domains*), que trabaja sobre problemas de programación con restricciones.

Esta es una práctica sobre el manejo de listas en Prolog tomando como excusa los sudokus. Por lo tanto, un Sudoku será una lista de longitud 81. Cada lugar en la lista se corresponde con una casilla del tablero 9x9. Se inicia con un tablero con números entre uno y nueve donde haya un número y con un punto en el resto. Vea el Ejemplo 1.

```
[., ., 3, ., 2, ., 7, ., .,  
5, ., ., ., ., ., 4, ., 3, .,  
. , ., 3, ., ., ., 2, 5, .,  
. , 5, ., 1, ., 6, ., .,  
. , ., 4, 8, ., 7, ., ., .,  
2, 3, 7, 6, ., 4, 8, ., .,  
. , 8, ., ., ., 2, ., 7, .,  
3, ., ., 4, ., ., 2, ., 8,  
. , ., 9, ., ., ., ., 6, .]
```

Ejemplo 1. Lista con un sudoku

En el Ejemplo 2 tenemos un sudoku típico.

		9	6			1		
8					1		9	
7								8
	3			6				
	4		1		9			5
9								
	8		9			5	4	
6			7	1				3
		5		8	4			9

Ejemplo 2. Sudoku típico

1. Objetivos

El objetivo de esta práctica es que el estudiante desarrolle habilidades en la manipulación de listas en Prolog, comprendiendo su importancia en la representación de problemas complejos mediante el sudoku. Además, se busca que practique la aplicación iterativa de reglas para la simplificación progresiva de la solución, explorando técnicas de búsqueda, sustitución y filtrado dentro de estructuras lineales. En general se pretende gestionar y manipular listas en Prolog tomando como caso de uso la representación y simplificación (parcial o total) de sudokus. Para lograrlo, se plantean:

1. Representación de un Sudoku como una lista lineal de longitud 81. Vea el Ejemplo 1.
2. Generación de una lista de posibilidades para cada casilla vacía (puntos).
Dado un sudoku, que recordemos, es una lista de longitud 81, se debe generar la lista de posibilidades para cada uno de los lugares del sudoku. Es decir, cada uno de los lugares ocupados por un punto debe sustituirse por una lista con todos los posibles números que pueden ir en ese lugar, mientras que los lugares ocupados inicialmente por dígitos no se tocan. La lista de posibilidades, por lo tanto, es una lista de la forma como se muestra en el Ejemplo3:

[[2,3,...],1,[3,4,...],..., [7,9]]

Ejemplo 3. Lista de posibilidades

Vea las listas de posibilidades en el sudoku del Ejemplo 4.

2345	25	9	6	23457	23578	2347	247
8	256	2346	2345	23457	1	23467	9
7	1256	12346	2345	23459	235	2346	2356
125	3	1278	2458	6	2578	12478	278
2	4	2678	1	237	9	23678	23678
9	12567	12678	23458	23457	23578	12346	23678
123	8	1237	9	23	236	5	4
6	29	24	7	1	25	28	28
123	127	5	23	8	4	1267	267

Ejemplo 4. Listas de posibilidades de un sudoku

3. Aplicación de reglas de simplificación para reducir las posibilidades de cada casilla:
Una vez generada la lista de posibilidades, se trata de simplificarla siguiendo una serie de reglas. Esta simplificación, en algunos casos, llevará a resolver el Sudoku, pero debe quedar claro que no siempre es el caso. Las cuatro reglas que hay que implementar son las siguientes:

Regla 0: Si hay un lugar donde solo cabe un número, lo escribimos en el lugar correspondiente y lo eliminamos de los lugares en los que aparezca dentro de los que son conflictivos.

Regla 1: Si hay un número que aparece en una sola de las listas que están en una fila, columna o cuadro, cambiamos la lista por ese número y borramos dicho número del resto de listas de la fila, columna o cuadro.

Regla 2: Si dos números aparecen solos en dos lugares distintos de una fila, columna o cuadro, los borramos del resto de lugares de la fila, columna o cuadro correspondiente.

Regla 3: Si en tres lugares de una fila, columna o cuadro solo aparecen tres números distintos, borramos los números de las restantes listas de la fila, columna o cuadro.

Nota 1: Estas reglas no se aplican de manera independiente unas de otras, sino que están relacionadas. Si en una fila se aplica la regla 2, por ejemplo, esto puede dar lugar a que se puedan reaplicar alguna de las reglas anteriores.

Nota 2: El estudiante debe proponer al menos 5 sudokus. Ver Apartado 3 Conjunto de ejemplos.

4. Comprobación parcial de la solución o resolución de algunos sudokus sencillos mediante la aplicación repetida de dichas reglas o aplicación de reglas adicionales producto del análisis de los resultados obtenidos al aplicar las 4 reglas enunciadas anteriormente.

2. Práctica a Entregar

La práctica consistirá en **dos partes principales**:

1. Programa Principal (main.pl)

- Donde se incluyan todos los predicados necesarios para:
 - Cargar la representación del Sudoku (lista de 81 elementos).
 - Generar las listas de posibilidades para cada casilla vacía.
 - Aplicar las reglas de simplificación (Regla 0, 1, 2 y 3).
 - Aplicación de reglas adicionales
 - métricas
 - Se recomienda sea modular por lo tanto el main.pl puede ser un script de entrada a la aplicación. Vea Formato del archivo en el Apartado 5.

2. Breve Informe (Informe.pdf)

- Explicando:
 - Objetivos alcanzados.
 - Descripción detallada de las reglas de simplificación y cómo están implementadas.
 - Posibles mejoras o extensiones (ver Sección 4).
 - Fuentes consultadas.
 - Distribución de tareas (en caso de trabajo en grupo).

3. Conjunto de ejemplos

Cada grupo debe proponer 5 Sudokus para resolver o simplificar. Estos Sudokus se deben entregar en la primera minuta de la práctica y servirán para:

- Probar la generación de posibilidades en cada casilla.
- Comprobar cómo funcionan las reglas de simplificación (Reglas 0, 1, 2 y 3).

- Analizar hasta dónde las reglas pueden simplificar cada Sudoku y, a partir de ahí, proponer mejoras (reglas adicionales) si fuera necesario.
- Medir cuántos de estos Sudokus se resuelven totalmente aplicando las reglas básicas y cuántos requieren mejoras adicionales.

Recomendaciones para la selección

1. Sudoku 1: Debe ser muy sencillo, de forma que se pueda resolver usando solo Regla 0 y Regla 1.
2. Sudoku 2: Requiere aplicar Regla 2 (pares exclusivos) o Regla 3 (tríos exclusivos) para avanzar.
3. Sudoku 3: De dificultad mayor, es posible que no se resuelva completamente con las reglas dadas. Sirve para comprobar la correcta iteración y ver hasta qué punto llega la simplificación.
4. Sudokus 4 y 5: Libres a elección del grupo. Pueden mostrar otras dificultades o servir para probar mejoras adicionales (por ejemplo, interfaz ASCII, nuevas reglas, etc.).

Qué se espera en el informe

- Listado de cada Sudoku en formato de lista de 81 casillas (o en una cuadrícula 9×9).
- Evolución de uno o dos ejemplos, mostrando cómo cambian las listas de posibilidades antes y después de aplicar cada regla.
- Resultados finales:
 - Si el Sudoku se resuelve por completo, mostrar la solución.
 - Si se queda a medias, indicar hasta dónde llegaron las reglas y proponer mejoras o reglas adicionales para completarlo.
- Métricas sobre cuántos Sudokus se resuelven aplicando únicamente las reglas básicas y cuántos requieren reglas o mejoras adicionales.

4. Mejoras del Programa

A. Posibles Mejoras

1. Interfaz ASCII: Mostrar el tablero antes y después de cada paso en un formato visual en texto.
2. Reglas Avanzadas: Incluir técnicas adicionales para resolver más casos.
3. Librería clpfd: Integrar clpfd para completar Sudokus no resueltos con reglas básicas y comparar eficiencia.
4. Iteración y Métricas: Automatizar la iteración de reglas y medir el número de iteraciones y casillas simplificadas.
5. Detección de Conflictos: Implementar verificación automática para identificar conflictos en filas, columnas y bloques.
6. Interfaz Interactiva: Permitir la entrada manual del Sudoku y mostrar el progreso paso a paso.

B. Metodología de Implementación de mejoras

1. Revisión teórica.
2. Propuesta de implementación.
3. Desarrollo del código en Prolog.
4. Pruebas y validación.
5. Análisis de resultados.
6. Conclusiones.
7. Documentación.

5. Detalles de la Entrega

- **Equipo:** De 3 personas, con al menos dos mejoras implementadas. Los equipos con más de tres personas deben adicionar una mejora por cada integrante adicional.
- **Cada equipo debe presentar resultados parciales cada dos semanas (minutas) donde especifique tarea realizada, resultados obtenidos y próximo paso.**

- **Formato del archivo:**

El archivo .zip debe tener la siguiente estructura:

- apellido1Apellido2Apellido3.Nombre_PLX.zip
 - ├── main.pl
 - ├── regla.pl
 - ├── metricas.pl
 - └── Informe.pdf
- Use camelCase para el nombre del archivo

- **Informe (15-20 páginas):**

- Portada.
- Resumen.
- Índice.
- Introducción.
- Objetivos.
- Desarrollo:
 - Descripción de la representación del Sudoku (lista de 81 casillas).
 - Generación de posibilidades para cada casilla vacía.
 - Aplicación de las reglas de simplificación (Reglas 0, 1, 2 y 3).
 - Mejoras implementadas y propuestas adicionales.
- Resultados:
 - Análisis de la evolución de ejemplos concretos.
 - Métricas: cuántos Sudokus se resuelven totalmente con las reglas básicas y cuántos requieren reglas adicionales.
- Conclusiones.
- Bibliografía (**OJO: es importante**)
- Anexos.

- **Entrega:**

- Plataforma: Campus Virtual de la UAH.
- Semana del PEC1

6. Aspectos de Evaluación y Rúbrica para la Práctica

- **Puntuación:** Cada práctica vale 100 puntos y se promedia la nota final aplicando el porcentaje correspondiente para prácticas de laboratorio.
- **Rúbrica:** se usa para evaluar todo el proceso de la práctica.
- **Proceso de Evaluación:** El profesor puede consultar sobre la práctica en cualquier momento. Al final de la práctica se realiza un proceso de defensa de la misma. El estudiante tiene derecho de defender su práctica.
- **Implementación de la práctica:** Debe estar operativa para evaluar el informe.
- **Minutas:** Se deben entregar al menos 2 minutos de la práctica. Estas contengan información sobre las tareas realizadas por el estudiante para que avance la práctica. Es individual. En ella se indica la tarea realizada por el estudiante, resultado obtenido y

la próxima tarea a realizar. La no entrega de minutas hace que el estudiante pierda la práctica. Si no recibes feedback de la minuta, se considera aprobada.

- Cronograma de Minutas: Primera minuta después de la explicación del profesor (siguiente semana obligatorio). Entrega cada dos semanas como máximo. La última minuta puede ser la entrega de la práctica final. En cada entrega se debe indicar la tarea a realizar.