

Problema del Caballo

Autores: Oscar Velasco y Luisa Salcedo

Docente: Edgar José Andrade



Logica para ciencias de la Computación
Proyecto final

May 24, 2019

Contenido

- 1 Introducción
- 2 Desarrollo
- 3 Resultados
- 4 Ejemplos
- 5 Conclusión

Contenido

- 1 Introducción
- 2 Desarrollo
- 3 Resultados
- 4 Ejemplos
- 5 Conclusión

- Problema principal
- Reglas del juego
- Reglas en lógica proposicional
- Representación del problema

Problema principal

En un tablero $n \times n$ se deben ubicar los números del 1 al número n^2 de tal forma que:

- 1) Ubicamos el número 1 en cualquier posición del tablero.
- 2) Luego, para ubicar el 2, debemos desplazarnos unicamente utilizando el movimiento en L del caballo de ajedrez, desde la posición donde ubicamos el 1.
- 3) Así sucesivamente ubicar los números hasta llegar a n^2 y así llenar todo el tablero, es decir encontrar la solución.

Ejemplo 4x4

1	2	3	4
5	6	7	8
9	11	10	13
12	14	15	16

Reglas del juego

- 1) Una vez ubicado el número 1, los demás números solo podrán ser ubicados desplazándose en L.
- 2) Debe haber exactamente un número en cada casilla del tablero.
- 3) Cada número debe estar solo una vez en el tablero.

Reglas en lógica proposicional

Regla 1: Solo se pueden ubicar los números usando el movimiento del caballo.

Representación del Problema

Nuestro tablero

A	B	C	D
E	F	G	H
I	J	K	L
M	N	P	Q

Reglas en lógica proposicional

Regla 1: Solo se pueden ubicar los números usando el movimiento del caballo.

Supongamos que se ubica 1 en A, luego tenemos dos casos para ubicar el 2, los cuales son G y J.

Representación del Problema

Nuestro tablero

1	B	C	D
E	F	2	H
I	2	K	L
M	N	P	Q

Reglas en lógica proposicional

Regla 1: Solo se pueden ubicar los números usando el movimiento del caballo.

Supongamos que se ubica 1 en A, luego tenemos dos casos para ubicar el 2, los cuales son G y J.

En lógica proposicional esto se representaría como: $A1 \rightarrow (G2 \vee J2)$

Reglas en lógica proposicional

1	B	3	D
E	F	2	3
3	2	K	L
M	3	P	3

Reglas en lógica proposicional

Regla 1: Solo se pueden ubicar los números usando el movimiento del caballo.

Ahora supongamos que ya anteriormente ubicamos el 1, y podemos ubicar 2 en A, es decir: $A2 \rightarrow (G3 \vee J3)$

Así sucesivamente con las demás casillas

Regla 2: Debe haber exactamente un número en cada casilla del tablero, es decir:

$$A1 \wedge \neg(-A2 \vee -A3 \vee \dots \vee -A16)$$

Reglas en lógica proposicional

Regla 2: Debe haber exactamente un número en cada casilla del tablero, es decir:

$$A1 \vdash \neg(-A2 \vee -A3 \vee \dots \vee -A16)$$

$$A2 \vdash \neg(-A1 \vee -A2 \vee \dots \vee -A16)$$

Reglas en lógica proposicional

Regla 2: Debe haber exactamente un número en cada casilla del tablero, es decir:

$$A1 \vdash \neg(-A2 \vee -A3 \vee \dots \vee -A16)$$

$$A2 \vdash \neg(-A1 \vee -A2 \vee \dots \vee -A16)$$

...

...

$$A16 \vdash \neg(-A1 \vee -A2 \vee \dots \vee -A15)$$

Reglas en lógica proposicional

Regla 2: Debe haber exactamente un número en cada casilla del tablero, es decir:

$A1 \wedge \neg(-A2 \vee -A3 \vee \dots \vee -A16)$

$A2 \wedge \neg(-A1 \vee -A2 \vee \dots \vee -A16)$

...

...

$A16 \wedge \neg(-A1 \vee -A2 \vee \dots \vee -A15)$

Del mismo modo con cada una de las casillas

Regla 3: Cada número debe estar exactamente una vez en el tablero, es decir:

$$A1 \wedge \neg(-B1 \vee -C1 \vee \dots \vee -Q1)$$

Reglas en lógica proposicional

Regla 3: Cada número debe estar exactamente una vez en el tablero, es decir:

$$A1 \vdash \neg(-B1 \vee -C1 \vee \dots \vee -Q1)$$

$$B1 \vdash \neg(-A1 \vee -C1 \vee \dots \vee -Q1)$$

Reglas en lógica proposicional

Regla 3: Cada número debe estar exactamente una vez en el tablero, es decir:

$$A1 \vdash \neg(-B1 \vee -C1 \vee \dots \vee -Q1)$$

$$B1 \vdash \neg(-A1 \vee -C1 \vee \dots \vee -Q1)$$

...

...

$$Q1 \vdash \neg(-A1 \vee -B1 \vee \dots \vee -P1)$$

Reglas en lógica proposicional

Regla 3: Cada número debe estar exactamente una vez en el tablero, es decir:

$$A1 \vdash \neg(-B1 \vee -C1 \vee \dots \vee -Q1)$$
$$B1 \vdash \neg(-A1 \vee -C1 \vee \dots \vee -Q1)$$

...

...

$$Q1 \vdash \neg(-A1 \vee -B1 \vee \dots \vee -P1)$$

Del mismo modo se debe realizar para todos los posibles números en las casillas

Contenido

- 1 Introducción
- 2 Desarrollo
- 3 Resultados
- 4 Ejemplos
- 5 Conclusión

Desarrollo del proyecto

- Algoritmos básicos
- ¿Cómo resolvimos el problema?

VI

Dada un árbol de una fórmula y una interpretación, esta retorna "True" si esta cumple las condiciones pautadas, en este caso, que se cumplan las 3 reglas.

Retorna "False" en otro caso.

String2Tree

Dada una fórmula escrita en polaca inversa, la retorna como un árbol.

InOrder

Dado un árbol, retorna su respectiva formula como una lista.

¿Cómo resolvimos el problema?

Tseitin

Recibe como parametro una fórmula y la retorna en FNC.

Forma Clausal

Dada una formula en FNC, la retorna en forma Clausal.

DPLL

Recibe como parametros un conjunto de clausulas, y una interpretación vacia, y retorna "Satisfacible" y su respectiva interpretación, o "Insatisfacible"

Contenido

- 1 Introducción
- 2 Desarrollo
- 3 Resultados**
- 4 Ejemplos
- 5 Conclusión

Resultados

DPLL tablero 3x3.

```
C:\Users\oscar\Desktop\Proyecto Logica Final\Tablero3x3>python3 proyecto_cuadro3x3.py
Importando paquetes...
Listo!

Letras de Tseitin: 2945

Tablero 3x3:

La formula es: Insatisfacible
Su interpretación es: {}
Por lo tanto este problema no tiene solución.
```

Resultados

DPLL tablero 4x4.

```
C:\Users\oscar\Desktop\Proyecto Logica Final\Tablero4x4>python3 proyecto_cuadro4x4.py
Importando paquetes...
Listo!

Letras de Tseitín: 16831

Tablero 4x4:

La formula es: Insatisfacible
Su interpretación es: {}
Por lo tanto este problema no tiene solución.
```

Contenido

- 1 Introducción
- 2 Desarrollo
- 3 Resultados
- 4 Ejemplos**
- 5 Conclusión

Ejemplos

3x3 con todas las casillas asignadas.

```
C:\Users\oscar\Desktop\Proyecto Logica Final>python3 regla3x3.py

Regla 1: 0
Regla 2: 1
Regla 3: 1

Tablero 3x3.

La interpretacion es :
A1 1
F2 1
G3 1
B4 1
I5 1
D6 1
C7 1
H8 1
E9 1

El valor de verdad de la regla es: 0
```

Ejemplos

3x3 con una casilla sin asignar.

```
C:\Users\oscar\Desktop\Proyecto Logica Final\Tablero4x4>CD..
C:\Users\oscar\Desktop\Proyecto Logica Final>python3 regla3x3.py

Regla 1: 0
Regla 2: 0
Regla 3: 0

Tablero 3x3.

La interpretacion es :
F2 1
G3 1
B4 1
I5 1
D6 1
C7 1
H8 1
E9 1

El valor de verdad de la regla es: 0

C:\Users\oscar\Desktop\Proyecto Logica Final>
```


Ejemplos

4x4 con todas las casillas asignadas.

```
C:\Users\oscar\Desktop\Proyecto Logica Final>python3 regla4x4.py

Regla 1: 0
Regla 2: 1
Regla 3: 1

Tablero 4x4.

La interpretacion es:
A01 1
B02 1
C03 1
D04 1
E05 1
F06 1
G07 1
H08 1
I09 1
J10 1
K11 1
L12 1
M13 1
N14 1
O15 1
P16 1

El valor de verdad es: 0
```

Ejemplos

4x4 con una casilla sin asignar.

```
C:\Users\oscar\Desktop\Proyecto Logica Final>python3 regla4x4.py

Regla 1: 0
Regla 2: 0
Regla 3: 0

Tablero 4x4.

La interpretacion es:
B02 1
C03 1
D04 1
E05 1
F06 1
G07 1
H08 1
I09 1
J10 1
K11 1
L12 1
M13 1
N14 1
P15 1
Q16 1

El valor de verdad es: 0
```

Ejemplos

5x5 con todas las casillas asignadas.

```
C:\Users\oscar\Desktop\Proyecto Logica Final>python3 regla5x5.py

Regla 1: 1
Regla 2: 1
Regla 3: 1

Tablero 5x5.

La interpretacion es:
R01 1
R02 1
R03 1
R04 1
R05 1
R06 1
R07 1
R08 1
R09 1
R10 1
R11 1
R12 1
R13 1
R14 1
R15 1
R16 1
R17 1
R18 1
R19 1
R20 1
R21 1
R22 1
R23 1
R24 1
R25 1

El valor de verdad es: 1
```

Ejemplos

5x5 con una casilla sin asignar.

```
C:\Users\oscar\Desktop\Proyecto Logica Final>python3 regla5x5.py

Regla 1: 0
Regla 2: 0
Regla 3: 0

Tablero 5x5.

La interpretacion es:
M02 1
Z03 1
S04 1
V05 1
L06 1
A07 1
H08 1
P09 1
N10 1
R11 1
F12 1
C13 1
J14 1
T15 1
M16 1
K17 1
B18 1
I19 1
U20 1
X21 1
Q22 1
G23 1
D24 1
M25 1

El valor de verdad es: 0

C:\Users\oscar\Desktop\Proyecto Logica Final>
```

Contenido

- 1 Introducción
- 2 Desarrollo
- 3 Resultados
- 4 Ejemplos
- 5 Conclusión**

Conclusión

Mientras mayor sea el n (tamaño del lado del tablero) mayor será el tiempo de ejecución del programa.

Esto se debe a que el número de letras proposicionales aumentan de forma exponencial dependiendo del tamaño del tablero.

Problema del Caballo

Autores: Oscar Velasco y Luisa Salcedo

Docente: Edgar José Andrade



Logica para ciencias de la Computación
Proyecto final

May 24, 2019