

# Pràctica ScalAT

Arnau Costa

Jan Puig

**Programació declarativa. Aplicacions**

EPS

Universitat de Girona

Gener 2025

# Contents

<b>1</b>	<b>Introducció</b>	<b>1</b>
<b>2</b>	<b>ScalAT</b>	<b>2</b>
2.1	Restriccions del Model SAT . . . . .	2
2.1.1	Restriccions Bàsiques . . . . .	2
2.1.2	Restriccions Implicades . . . . .	3
2.1.3	Trencament de Simetries . . . . .	3
<b>3</b>	<b>Execucio</b>	<b>5</b>
<b>4</b>	<b>Conclusions</b>	<b>8</b>

# Chapter 1

## Introducció

Aquest treball té com a objectiu aprofundir en tècniques avançades de resolució de problemes mitjançant programació per restriccions (CP) i satisfactibilitat booleana (SAT). La pràctica es divideix en dues parts diferenciades, cadascuna centrada en un enfocament i metodologia específica, en aquest PDF es presenta la part de CP.

A aquesta segona part, s'utilitzarà ScalAT, un SAT solver implementat en Scala, per desenvolupar un solucionador per al joc del Buscamines.

# Chapter 2

## ScalAT

### 2.1 Restriccions del Model SAT

En aquest apartat es descriuen les restriccions implementades en el model SAT per resoldre el problema del BuscaMines, seguint el viewpoint basat en variables duals per casella.

#### 2.1.1 Restriccions Bàsiques

Les restriccions bàsiques garanteixen que el model representa correctament les regles del joc. Aquestes són:

1. **Restricció de "Exactly-One" per casella:** Cada casella pot tenir exactament un dels dos estats possibles:

- Conté una mina.
- No conté una mina.

Aquesta restricció es codifica com:

$$\text{Exactly-One}(v_{ij}^{\text{mine}}, v_{ij}^{\text{noMine}})$$

per a cada casella  $(i, j)$ .

2. **Restriccions locals per a números (pistes):** Si una casella conté un número  $n$ , exactament  $n$  de les caselles adjacents han de ser mines. Aquesta restricció es

codifica com:

$$\text{Exactly-K}(\text{adj}(i, j), n)$$

on  $\text{adj}(i, j)$  són les variables de les caselles adjacents a  $(i, j)$ .

3. **Restricció global del nombre total de mines:** Si es coneix el nombre total de mines  $n_{\text{mines}}$ , aquest valor ha de coincidir amb la suma de totes les caselles marcades com a mines. Això es codifica com:

$$\text{Exactly-K}(\{v_{ij}^{\text{mine}} \mid \forall(i, j)\}, n_{\text{mines}})$$

### 2.1.2 Restriccions Implicades

Les restriccions implicades no formen part explícita del problema, però s'introdueixen per optimitzar el model i millorar el rendiment:

1. **Caselles amb valor 0:** Si una casella conté el número 0, cap de les caselles adjacents pot contenir una mina. Aquesta restricció es codifica com:

$$\text{At-Most-0}(\text{adj}(i, j))$$

2. **Caselles amb el màxim valor de mines:** Si una casella conté un número igual al nombre màxim de caselles adjacents, totes aquestes han de ser mines. Aquesta restricció es codifica com:

$$\text{Exactly-K}(\text{adj}(i, j), |\text{adj}(i, j)|)$$

3. **Adjacents complets:** Si el nombre de mines d'una casella és igual al nombre de caselles adjacents no resoltes, totes aquestes caselles han de ser mines.

### 2.1.3 Trencament de Simetries

Tot i que el problema del BuscaMines no presenta simetries evidents en la seva formulació, s'han introduït algunes optimitzacions per evitar solucions equivalents:

1. **Ordre lexicogràfic en les variables:** Es defineix un ordre de prioritats entre les variables per evitar configuracions equivalents. Per exemple, entre dues caselles adjacents  $(i, j)$  i  $(k, l)$ , es pot imposar:

$$\neg v_{ij}^{\text{mine}} \vee v_{kl}^{\text{mine}}$$

per assegurar que  $v_{ij}^{\text{mine}}$  és prioritari respecte a  $v_{kl}^{\text{mine}}$ .

2. **Regió idèntica:** Si dues regions del tauler tenen el mateix patró de pistes i caselles buides, s'imposa una ordre per garantir que una regió es resolgui primer.
3. **Caselles amb el mateix número:** Si dues caselles tenen el mateix valor i adjacències idèntiques, es poden considerar com una única entitat per reduir el nombre de variables i clàusules.

Aquestes optimitzacions redueixen la mida de l'espai de cerca i acceleren la resolució del problema mitjançant el SAT solver.

## Chapter 3

# Execució

### Abast d'execució

L'execució del codi ha sigut satisfactòria. El codi resol de forma correcta les diverses matrius de mines que s'han passat i dona un missatge en cas contrari.

### Nota sobre l'anàlisi de temps d'execució

Durant el desenvolupament del nostre programa, hem observat que aquest resol tots els problemes plantejats, incloent el cas més gran de 100x100, de manera pràcticament instantània. Aquest comportament ens ha generat dubtes sobre si hem entès correctament l'enunciat del problema, ja que esperàvem que els casos més grans requerissin temps d'execució significativament més elevats.

Com a conseqüència, no hem pogut realitzar una taula comparativa de temps d'execució per a diferents dimensions del problema, ja que els resultats són uniformement immediats. Això podria indicar que el problema tal com l'hem interpretat està sent resol't d'una manera més eficient del que s'esperava, o bé que estem tractant un cas diferent de l'esperat.

2	4		4	3	3		1	1		1		2		2	1	1	0		
	5						3		3	3	1	3		4		1		2	4
1			4	2			1					1		4			3	6	
	3	3	3				4	4	6						2		4	7	
1				1							2		5	3		3			
1			2				3	4		1		2				5			
	3	3	4				3	2		1	1			2	2		3		1
1							3				1	1			2		2	1	
		4					5	3		4	4				0				
		0				4	1	1	3				2	1			1	1	
	3		5			3	2	1	2			3		2					3
4				3				1	4			3		3					2
2	3						2	1				3		2	1	1			1

FIGURE 3.1: Imatge del taulell 129 sense resoldre

```
Solution found:
```

X	X	o	X	X	X	X	X	X	X	o	X	X	o	X	X	X	X	X	o	X
o	o	o	o	o	o	X	o	X	X	X	X	o	o	X	o	X	X	X	X	
X	X	o	X	X	X	X	X	X	X	X	o	X	X	X	X	o	X	o	X	
X	o	X	X	o	X	o	X	o	o	X	X	X	X	o	X	X	o	X	o	
X	X	o	X	X	X	X	X	X	o	X	X	o	o	X	X	X	o	X	o	
X	o	X	X	o	X	o	o	o	X	o	X	X	X	X	o	X	o	o	o	
X	X	o	X	X	X	X	X	X	X	X	o	X	o	X	o	X	o	X	o	
X	X	X	o	X	o	X	X	X	X	X	X	X	X	X	o	o	X	o	X	
X	o	o	X	o	o	X	o	X	o	X	X	X	X	X	X	X	X	X	X	
X	X	X	o	o	X	X	X	X	o	X	o	X	o	X	X	X	X	X	X	
X	X	X	o	o	X	o	X	X	X	X	o	X	X	X	X	X	X	X	X	
o	X	X	X	o	X	X	X	X	X	o	X	X	X	o	X	o	X	o	X	
o	X	o	o	X	o	X	o	X	o	X	o	o	X	o	X	o	X	X	X	
X	o	X	X	X	X	o	X	X	X	X	o	X	X	X	X	X	X	X	X	

FIGURE 3.2: Imatge del taulell 129 resolt per el programa

## Ús de l'Aplicatiu

### Execució amb un fitxer d'entrada

Per executar els aplicatius `BuscaMinesLog` o `BuscaMinesQuad`, cal proporcionar com a paràmetre el nom del fitxer que conté les dades d'entrada. Això es pot fer directament des de la línia de comandes, indicant el nom del fitxer en el moment de l'execució.

### Execució a IntelliJ IDEA

Si es fa ús de l'entorn de desenvolupament IntelliJ IDEA, el nom del fitxer es pot especificar a l'apartat *Program Arguments* dins del menú *Build and Run* de la configuració d'execució del programa.



## Fitxers Disponibles

Els fitxers d'entrada que es poden utilitzar són els següents:

- `37.txt`: Equivalent a <https://www.janko.at/Raetsel/Minesweeper/037.a.htm>.
- `129.txt`: Equivalent a <https://www.janko.at/Raetsel/Minesweeper/129.a.htm>.
- `292.txt`: Equivalent a <https://www.janko.at/Raetsel/Minesweeper/292.a.htm>.
- `custom.txt`: Fitxer generat manualment que conté una graella de 100x100. Aquest fitxer ha estat creat específicament per a proves personalitzades.

És imprescindible assegurar-se que el fitxer especificat com a paràmetre és accessible i compleix amb el format esperat per l'aplicatiu.

## Chapter 4

# Conclusions

Com a conclusió, i en relació amb la nostra incertesa sobre si el temps de resolució ha d'escalar més o no, després de reflexionar-hi, considerem que, dins d'un buscamines, els problemes són més localitzats. Un problema en una part de la matriu no necessàriament afecta una altra part llunyana. Per tant, creiem que el nostre codi és correcte.