

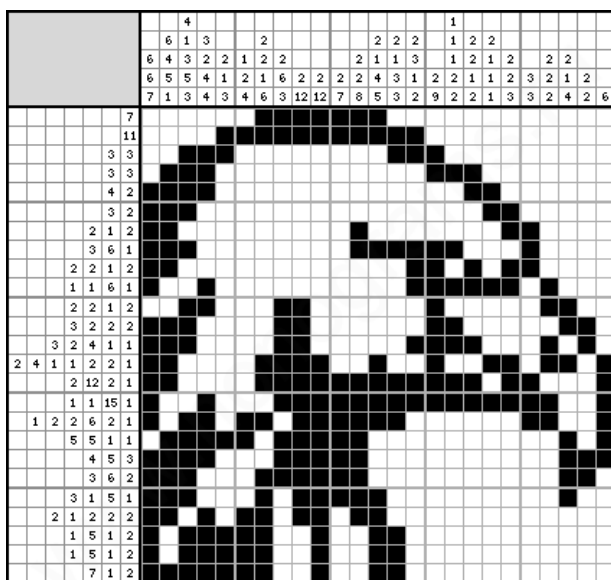
## Term Project Milestone3 實作報告

林怡秀 0556017

## SAT Application: Using my YaSat to solve Nonogram

## 1. Introduction:

Nonogram puzzle 如下圖，每行(列)puzzle 的最上(右)方的數字分別表示此圖形的限制條件，舉下圖第一行(column)為例：限制條件為 6、6、7，代表此行必出現的連續黑格數目，而其中每一段連續黑格間必需要空至少一個白格。Nonogram 已在 1996 年被證實為 NP-complete 問題 [1]。



## 2. Encoding Method

我嘗試使用以下兩種不同的 encoding 方式，第一種方法很直覺，但變數量與 clause 量相對較多，第二種方式參考論文[2]實作，效率提高不少。

### (1) First Method

將題目中每格對應為一個變數，計算出每行(列)的所有可能著色，每行(列)會形成 DNF 的 Boolean expression，再將每行(列)and 起來便完成 encode。過程中須將 DNF 轉成 CNF 形式且為了不讓 clause 量 exponential 成長，需

引進新的變數。以下舉例說明。

		x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
2	6										
2	6										
2	6										

上圖為 Nonogram puzzle 中的其中一列，當限制條件為 2,6 時會有 3 種不同可能，對應的 Boolean expression 為:

$$(x1 \wedge x2 \wedge \sim x3 \wedge x4 \wedge x5 \wedge x6 \wedge x7 \wedge x8 \wedge x9 \wedge \sim x10) \vee$$

$$(x1 \wedge x2 \wedge \sim x3 \wedge \sim x4 \wedge x5 \wedge x6 \wedge x7 \wedge x8 \wedge x9 \wedge x10) \vee$$

$$(\sim x1 \wedge x2 \wedge x3 \wedge \sim x4 \wedge x5 \wedge x6 \wedge x7 \wedge x8 \wedge x9 \wedge x10) \vee$$

以此類推，每行(列)皆可 encode 成上述 DNF 形式，為了使上述 DNF 轉成 CNF，需引進新變數(y1, y2, y3)，以下 CNF 為上述 DNF 轉換後：

$$(y1 \vee y2 \vee y3) \wedge$$

$$(\sim y1 \vee x1) \wedge (\sim y1 \vee x2) \wedge (\sim y1 \vee \sim x3) \wedge (\sim y1 \vee x4) \wedge (\sim y1 \vee x5) \wedge$$

$$(\sim y1 \vee x6) \wedge (\sim y1 \vee x7) \wedge (\sim y1 \vee x8) \wedge (\sim y1 \vee x9) \wedge (\sim y1 \vee \sim x10) \wedge$$

$$(\sim y2 \vee x1) \wedge (\sim y2 \vee x2) \wedge (\sim y2 \vee \sim x3) \wedge (\sim y2 \vee \sim x4) \wedge (\sim y2 \vee x5) \wedge$$

$$(\sim y2 \vee x6) \wedge (\sim y2 \vee x7) \wedge (\sim y2 \vee x8) \wedge (\sim y2 \vee x9) \wedge (\sim y2 \vee x10) \wedge$$

$$(\sim y3 \vee \sim x1) \wedge (\sim y3 \vee x2) \wedge (\sim y3 \vee x3) \wedge (\sim y3 \vee \sim x4) \wedge (\sim y3 \vee x5) \wedge$$

$$(\sim y3 \vee x6) \wedge (\sim y3 \vee x7) \wedge (\sim y3 \vee x8) \wedge (\sim y3 \vee x9) \wedge (\sim y3 \vee x10)$$

以上述轉換方法實作每行(列)再 AND 所有 CNF 便完成 encoding。

## (2) Second Method

第二種 encoding 方法將每個連續黑格的起始位置設變數，將每個整數變數利用 ordering encoding 轉成布林變數，再分別對 Nonogram 的以下四種特性做編碼：

(a) 起始位置介於 1~n (n: puzzle 大小)

(b) 在一行(列)中下一個黑格起始位置必大於上一個黑格起始位置+數字限制+1(中間需空白格)

(c) 黑格起始位置加上此限制數字範圍著黑

(d) 在一行(列)中下一個黑格起始位置與上一個連續黑格尾端間為白

以下取 puzzle 中的其中一列舉例說明，此列的限制數字為 $\langle 3, 1, 2 \rangle$ ：

令  $X = \langle x_1, x_2, x_3, \dots, x_{10} \rangle$ 、 $\bar{X} = \langle \sim x_1, \sim x_2, \sim x_3, \dots, \sim x_{10} \rangle$ 、 $U_1, U_2, U_3$  分別為此列連續黑格的起始位置：

$$U_1 = \langle a_1, a_2, a_3, \dots, a_{10} \rangle$$

$$U_2 = \langle b_1, b_2, b_3, \dots, b_{10} \rangle$$

$$U_3 = \langle c_1, c_2, c_3, \dots, c_{10} \rangle$$

其中， $a_i$  為 1 表示  $U_1 \geq i$  為 True， $U_2, U_3$  同定義，舉例：

$U_1 = \langle 1, 1, 1, 0, 0, 0, 0, 0, 0, 0 \rangle$  表示： $U_1 \geq 1, U_1 \geq 2, U_1 \geq 3, U_1 < 4, U_1 < 5 \dots$ ，故可得知  $U_1 = 3$ 。又  $U_1^{+5}$  表示  $U_1 + 5 = \langle 1, 1, 1, 1, 1, 1, 1, 1, 0, 0 \rangle = 3 + 5 = 8$ ， $U_1^{-2}$  表示  $U_1 - 2 = \langle 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 \rangle = 3 - 2 = 1$

根據上述四個特性對 Nonogram 進行 encoding：

(a)  $U_1, U_2, U_3$  介於 1~10： $\text{unary}(U_1, [1, 10]), \text{unary}(U_2, [1, 10]), \text{unary}(U_3, [1, 10])$

(b)  $U_1 + \text{Length} \leq U_2$ ： $\text{leq}(U_1^{+3}, U_2), \text{leq}(U_2^{+1}, U_3), \text{leq}(U_3^{+1}, 10)$

(c)  $X=1$  範圍： $\text{block}(U_1^{-1}, U_1^{+2}, X), \text{block}(U_2^{-1}, U_2^{+0}, X), \text{block}(U_3^{-1}, U_3^{+1}, X)$

(d)  $X=0$  範圍： $\text{block}(0, U_1, \bar{X}), \text{block}(U_1^{+3}, U_2, \bar{X}), \text{block}(U_2^{+1}, U_3, \bar{X}),$   
 $\text{block}(U_3^{+2}, 10, \bar{X})$

其中 unary, leq, block 定義如下：

$$\text{unary}(\langle x_1, \dots, x_n \rangle, [a, b]) = \bigwedge_{i=1}^n (x_{i-1} \leftarrow x_i) \wedge x_a \wedge \neg x_{b+1}$$

$$\text{block}(U_1, U_2, \langle x_1, \dots, x_n \rangle) = \bigwedge_{i=1}^n (\neg U_1(i) \wedge U_2(i)) \rightarrow x_i$$

$$\text{leq}(\langle x_1, \dots, x_n \rangle, \langle y_1, \dots, y_n \rangle) = \bigwedge_{i=1}^n x_i \rightarrow y_i$$

### 3. Experiment

以下比較 2 種 encoding method，包括執行時間與產生變數和 clause 的量

Method Test case	Method1 run time (second)	Method2 run time (second)
5*5 puzzle1	0.002	0.014
10*10 puzzle1	0.104	0.002

25*25 puzzle1	$\infty$	3.802
25*25 puzzle2	$\infty$	4.790
25*25 puzzle3	$\infty$	3.633
25*25 puzzle4	$\infty$	3.833
25*25 puzzle5	$\infty$	26.982

<b>Method</b> <b>Test case</b>	<b>Method1</b> <b>Variable/Clause</b>	<b>Method2</b> <b>Variable/Clause</b>
5*5 puzzle1	60/180	116/311
10*10 puzzle1	377/2780	472/1349
25*25 puzzle1	180230/4490150	9508/34234
25*25 puzzle2	245342/6117950	9562/34434
25*25 puzzle3	206815/5154775	9562/34433
25*25 puzzle4	261970/6533650	9373/33686
25*25 puzzle5	209123/5212475	9535/34342

p.s. 以上測資皆放在上傳作業的 testcase 資料夾中(格式請參見測資檔案，25\*25 測資來自 TAAI 2016 Nonogram 比賽前 5 題)，另外若需跑測資需依大小改程式中的 LENGTH、ROW、COL、CONSTRAINTn(method1)變數(目前預設 25\*25)。

執行方式： g++ sat\_encode1.cpp/sat\_encode2.cpp

./a.out <testcase\_name>.txt

puzzle 答案會產生於 puzzleans.txt

#### 4. Difficulty

一開始實作第一種非常直覺的 encoding method 效果不彰，但一直想不到更好的 encoding 方式，在網路上找了一些資料，其實 Nonogram 的解法很多，真正使用 sat solver 解的只是其中一部分，很幸運地找到篇論文提到如何 encode Nonogram，但論文中仍有未考量的部分，例如：最後一個連續黑格尾端不能超出 puzzle 大小、最後一個連續黑格尾端至 puzzle 大小需留白等。另外我覺得使用 ordering encoding 雖然比較不直覺，但卻不失為一個好方法。

## 5. Reference:

- [1] Ueda, Nobuhisa; Nagao, Tadaaki (1996), NP-completeness results for NONOGRAM via Parsimonious Reductions, TR96-0008, Technical Report, Department of Computer Science, Tokyo Institute of Technology, CiteSeerX 10.1.1.57.5277
- [2] Amit Metodi, Michael Codish, Vitaly Lagoon, Peter J. Stuckey (2011), Boolean Equi-propagation for Optimized SAT Encoding
- [3] Naoyuki Tamura, Tomoya Tanjo, and Mutsunori Banbara, Solving Constraint Satisfaction Problems with SAT Technology