



PRESENTATION - PROGRESS II

Artificial Intelligence



MEMBER OF THE GROUP

- ❑ **52100871 – Nguyễn Quốc Anh**
- ❑ **52100947 – Võ Phú Vinh**
- ❑ **52100944 – Nguyễn Vũ Tường**
- ❑ **52100674 – Trần Thị Vẹn**

BẢNG NỘI DUNG

01

BÀI TOÁN & GIẢI PHÁP

Giới thiệu về bài toán

02

THUẬT TOÁN

Phân tích thuật toán

03

GIẢI THÍCH

Giải thích rõ hơn về thuật toán

04

KẾT QUẢ & ĐÁNH GIÁ

Hiển thị kết quả và các đánh giá về độ hiệu quả



01

BÀI TOÁN & GIẢI PHÁP

BÀI TOÁN & GIẢI PHÁP

01 BÀI TOÁN

- Matrix kích thước $m \times n$, mỗi ô chứa số nguyên không âm hoặc trống.
- Mỗi ô có 9 ô "liền kề", bao gồm chính nó và 8 ô xung quanh.
- Người chơi sẽ tô màu ô đỏ hoặc xanh lá cây sao cho số lượng ô xanh lá cây liền kề với một ô bằng chính số trong ô đó.
- Không có ràng buộc gì đối với các ô trống.

02 GIẢI PHÁP

- Xây dựng CNF với PySAT
- Truyền CNF vào Glucose3
- Tạo ra các điều kiện
- Thêm các điều kiện này vào biểu thức CNF
- Sử dụng SAT solver để tìm một phép tô màu thỏa mãn
- Sử dụng module colorama để in ra ma trận đã được tô màu



02

THUẬT TOÁN

Ý TƯỞNG CHÍNH ĐỂ THỰC HIỆN

01 → Tìm các tập hợp mà hàm logic của một ô trên ma trận có giá trị sai.

02 → Chuyển các tập hợp trên thành các điều kiện logic dưới dạng DNF và phủ định để có thể biểu diễn bằng CNF.

03 → Thêm các điều kiện này vào trong SAT solver để giải quyết bài toán tô màu đồ thị.

MINH HỌA THUẬT TOÁN

Construction of the conjunctive normal form:

Find the sets on which the function takes a false value:

$\{1, 1, 0, 1\} \{1, 1, 1, 0\} \{1, 1, 1, 1\}$

Let us match the found sets with elementary disjunctions for all variables, and if a variable in the set takes value 1, it will be written with negation:

$D_1: \{1, 1, 0, 1\} \rightarrow \neg X_1 \vee \neg X_2 \vee X_3 \vee \neg X_4$

$D_2: \{1, 1, 1, 0\} \rightarrow \neg X_1 \vee \neg X_2 \vee \neg X_3 \vee X_4$

$D_3: \{1, 1, 1, 1\} \rightarrow \neg X_1 \vee \neg X_2 \vee \neg X_3 \vee \neg X_4$

Combine the disjunctions using the AND operation and obtain a perfect conjunctive normal form:

$D_1 \wedge D_2 \wedge D_3 = (\neg X_1 \vee \neg X_2 \vee X_3 \vee \neg X_4) \wedge (\neg X_1 \vee \neg X_2 \vee \neg X_3 \vee X_4) \wedge (\neg X_1 \vee \neg X_2 \vee \neg X_3 \vee \neg X_4)$



03

GIẢI THÍCH

GIẢI THÍCH CHI TIẾT

- ✓ Để tạo ra dạng CNF, đầu tiên ta cần tìm tập hợp các phép toán hàm mà có giá trị sai (trong trường hợp này là false), sau đó chuyển đổi chúng thành các điều kiện logic dưới dạng DNF và phủ định để có thể biểu diễn bằng CNF.
- ✓ Trong ví dụ này, các tập hợp được tìm thấy là $\{1, 1, 0, 1\}$, $\{1, 1, 1, 0\}$, và $\{1, 1, 1, 1\}$. Chúng được kết hợp với các phép toán hàm tuyển tính để tạo thành các mệnh đề dạng disjunctive normal form (DNF) và phủ định.
- ✓ Kết quả thu được là một dạng CNF hoàn hảo.



04

KẾT QUẢ & ĐÁNH GIÁ

KẾT QUẢ

01 TEST CASE:

10	10								
.	2	3	.	.	0
.	.	.	.	3	.	2	.	.	6
.	.	5	.	5	3	.	5	7	4
.	4	.	5	.	5	.	6	.	3
.	.	4	.	5	.	6	.	.	3
.	.	.	2	.	5
4	.	1	.	.	.	1	1	.	.
4	.	1	.	.	.	1	.	4	.
.	.	.	.	6	4
.	4	4	4	.	.

02 RESULT:

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

x
x x
x

ĐÁNH GIÁ

1. Chức năng của code này là giải quyết một bài toán bằng cách sử dụng phương pháp mã hóa CNF và tìm kiếm SAT bằng Glucose3.
2. Các bước thực hiện của code bao gồm:
 - Đọc ma trận từ file input.txt.
 - Sử dụng phương pháp mã hóa CNF để biến bài toán này thành một bài toán SAT.
 - Sử dụng Glucose3 để giải quyết bài toán SAT.

ĐÁNH GIÁ

ƯU ĐIỂM

- Ưu điểm của phương pháp mã hóa CNF là giúp chuyển bài toán sang bài toán SAT một cách tự động, tiết kiệm thời gian và công sức so với việc giải trực tiếp bài toán Sudoku.
- Ngoài ra, Glucose3 là một trong những công cụ giải quyết SAT hiệu quả, giúp tìm kiếm lời giải một cách nhanh chóng.

HẠN CHẾ

- Phương pháp mã hóa CNF có thể tạo ra rất nhiều biến và ràng buộc, do đó cần phải đảm bảo rằng Glucose3 có thể giải quyết bài toán với số biến và ràng buộc lớn.
- Thời gian giải quyết của Glucose3 có thể phụ thuộc vào độ phức tạp của bài toán SAT, do đó có thể mất thời gian để tìm kiếm lời giải đối với các bài toán có độ phức tạp lớn.

TÀI LIỆU THAM KHẢO

- [1]. J. P. Marques-Silva, I. Lynce, and S. Malik, “Conflict-driven clause learning SAT solvers,” in Handbook of Satisfiability, A. Biere, M. Heule, H. van Maaren, and T. Walsh, Eds. IOS Press, 2009, pp. 131–153. DOI: 10.3233/978-1-58603-929-5-131
- [2]. M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik, “Chaff: Engineering an efficient SAT solver,” in Proceedings of the 38th Design Automation Conference, 2001, pp. 530–535. DOI: 10.1145/378239.379017
- [3]. D. E. Mitchell, B. Selman, and H. J. Levesque, “Hard and easy distributions of SAT problems,” in Proceedings of the 10th National Conference on Artificial Intelligence, 1992, pp. 459–465. <https://www.aaai.org/Papers/AAAI/1992/AAAI92-076.pdf>
- [4]. F. Bacchus and T. Walsh, “Modelling CSPs with relaxed quantifiers,” in Proceedings of the 16th National Conference on Artificial Intelligence, 1999, pp. 109–116. <https://www.aaai.org/Papers/AAAI/1999/AAAI99-015.pdf>

THANKS!

