

# Constraint Satisfaction Problems (CSPs) | 25/06/25 คณณน

ปัญหาที่กำหนด ตัวแปร, โดเมน, และ ข้อจำกัด เช่น Sudoku, Map-coloring  
ซึ่งลดตัวแปรลงในขั้นตอนต้นๆ

## องค์ประกอบของ CSPs

Variables : ตัวแปร

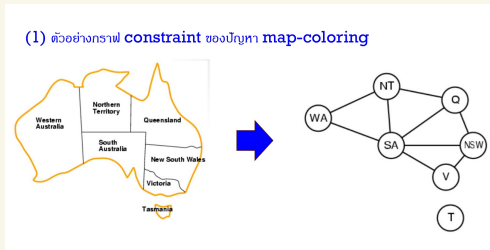
Domain : ชุดค่าที่เลือกได้ของตัวแปร

Constraints : ข้อจำกัดที่กำหนด

## ชนิดของ Constraints (ข้อจำกัด)

- Unary ได้เกี่ยวกับตัวแปรตัวเดียว เช่น  $SA \neq \text{green}$
- Binary ได้เกี่ยวกับตัวแปร 2 ตัว เช่น  $SA \neq WA$
- Higher-order ได้เกี่ยวกับ 3 ตัวแปร ขึ้นไป
- Soft เชื่อมโยงแบบไม่บังคับ

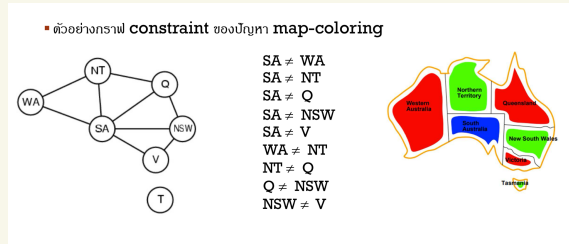
## Constraint Graph แปลงเป็นกราฟดูง่ายขึ้น



จากตัวอย่างปัญหา Map Coloring สามารถแปลงปัญหาให้อยู่ในรูปแบบของ Constraint Graph ได้ดังรูป โดยที่แต่ละวงที่เชื่อมถึงกันจะแทนด้วยขอบเขต เช่น Western Australia จะกลายเป็นโหนด WA

จากนั้นสามารถเขียนสมการระหว่างโหนดเพื่อแสดงความสัมพันธ์หรือข้อกำหนด โดยจากภาพจะเห็นได้ว่า จะมี Binary Constraint อยู่ 9 เงื่อนไขคือ

- ถ้าโหนด SA เชื่อมกับโหนด WA:  $SA \neq WA$
- ถ้าโหนด SA เชื่อมกับโหนด NT:  $SA \neq NT$
- ถ้าโหนด SA เชื่อมกับโหนด Q:  $SA \neq Q$
- ถ้าโหนด SA เชื่อมกับโหนด NSW:  $SA \neq NSW$
- ถ้าโหนด SA เชื่อมกับโหนด V:  $SA \neq V$
- ถ้าโหนด WA เชื่อมกับโหนด NT:  $WA \neq NT$
- ถ้าโหนด NT เชื่อมกับโหนด Q:  $NT \neq Q$
- ถ้าโหนด Q เชื่อมกับโหนด NSW:  $Q \neq NSW$
- ถ้าโหนด NSW เชื่อมกับโหนด V:  $NSW \neq V$



## วิธีแก้ปัญหา CSPs

1. Systematic Search Algorithm ไล่ผ่านคำตอบจนกว่ามีระบบ

เช่น - Generate and Test (GT) → เตาสุ่มคำตอบจนได้คำตอบที่ขึ้นแล้ว

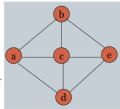
เป็นเทคนิคแบบ Brute Force (ห่วย)

- Backtracking (BT) → DFS + ย้อนกลับเมื่อเจอทางตัน

ตัวอย่างโจทย์:

## CLASS ASSIGNMENT

ข้อ (1) กำหนดให้ constraint graph ของ CSP ปัญหาหนึ่งมีลักษณะเป็นดังรูปที่กำหนดให้  
 และ:  
 กำหนดโดเมน (domain) ของตัวแปรแต่ละตัวตามเงื่อนไขคือ  $\{1,2,3,4\}$   
 เมื่อกำหนด unary constraint ดังต่อไปนี้  
 - variable "a"  $\neq 3, 4$   
 - variable "b"  $\neq 4$



ข้อ (2) กำหนดให้ constraint graph ของ CSP ปัญหาหนึ่งมีลักษณะเป็นดังรูปที่กำหนดให้  
 และ:  
 กำหนดโดเมน (domain) ของตัวแปรแต่ละตัวตามเงื่อนไขคือ  $\{1,2,3,4\}$   
 เมื่อกำหนด unary constraint ดังต่อไปนี้  
 - variable "a"  $\neq 3, 4$   
 - variable "b"  $\neq 4$

1.1 5 constraints a, b, c, d, e

1.2 มี 8 constraints

a  $\neq$  b, a  $\neq$  c, a  $\neq$  d, b  $\neq$  c, b  $\neq$  e,

c  $\neq$  e, c  $\neq$  d, d  $\neq$  e

## CLASS ASSIGNMENT

ข้อ (2) Let's consider a small part (wheel installation) of the car assembly consisting of 15 tasks:  
 • Install axles (front and back) - requires 10 mins to install  
 • Affix all four wheels (right and left, front and back) - takes 1 mins  
 • Tighten nuts for each wheel - takes 2 mins  
 • Affix hubcaps and - requires 1 mins  
 • Inspect the final assembly - takes 3 mins  
 And get the whole assembly done in 30 mins



ข้อ (2)

We can represent the tasks with 15 variables:  
 $X = \{ \text{Axle}_F, \text{Axle}_B, \text{Wheel}_{LF}, \text{Wheel}_{LF}, \text{Wheel}_{LB}, \text{Wheel}_{LB}, \text{Nuts}_{LF}, \text{Nuts}_{LF}, \text{Nuts}_{LB}, \text{Nuts}_{LB}, \text{Cap}_{LF}, \text{Cap}_{LF}, \text{Cap}_{LB}, \text{Cap}_{LB}, \text{Inspect} \}$

Precedence constraints  $\Rightarrow$  whenever a task  $T_i$  must occur before task  $T_j$  and task  $T_j$  takes duration  $d_j$  to complete, then we can represent the constraint as

$$T_i + d_i \leq T_j$$

หรือ

$$\text{Axle}_F + 10 \leq \text{Wheel}_{LF}$$

$$\text{Axle}_F + 10 \leq \text{Wheel}_{LB}$$

2.1 กำหนด constraints ทั้งหมดของปัญหานี้  
 2.2 domain ของตัวแปรในปัญหานี้คืออะไร

$$\text{Axle}_F, \text{Axle}_B \rightarrow 10 \text{ min}$$

$$\text{Wheel}_{LF}, \text{Wheel}_{LB}, \text{Wheel}_{LF}, \text{Wheel}_{LB} \rightarrow 1 \text{ min}$$

$$\text{Nuts}_{LF}, \text{Nuts}_{LB}, \text{Nuts}_{LF}, \text{Nuts}_{LB} \rightarrow 2 \text{ min}$$

$$\text{Cap}_{LF}, \text{Cap}_{LB}, \text{Cap}_{LF}, \text{Cap}_{LB} \rightarrow 1 \text{ min}$$

$$\text{Inspect} \rightarrow 3 \text{ min}$$

งาน  $T_1$  ต้องเสร็จก่อน  $T_2$

1. Affix all four wheel

$$\text{Axle}_F + 10 \leq \text{Wheel}_{LF}$$

$$\text{Axle}_F + 10 \leq \text{Wheel}_{LB}$$

$$\text{Axle}_B + 10 \leq \text{Wheel}_{LF}$$

$$\text{Axle}_B + 10 \leq \text{Wheel}_{LB}$$

2. Tighten nuts

$$\text{Wheel}_{LF} + 1 \leq \text{Nuts}_{LF}$$

$$\text{Wheel}_{LB} + 1 \leq \text{Nuts}_{LB}$$

$$\text{Wheel}_{RF} + 1 \leq \text{Nuts}_{RF}$$

$$\text{Wheel}_{RB} + 1 \leq \text{Nuts}_{RB}$$

3. Affix hubcaps

$$\text{Nuts}_{LF} + 2 \leq \text{Cap}_{LF}$$

$$\text{Nuts}_{LB} + 2 \leq \text{Cap}_{LB}$$

$$\text{Nuts}_{RF} + 2 \leq \text{Cap}_{RF}$$

$$\text{Nuts}_{RB} + 2 \leq \text{Cap}_{RB}$$

4. Inspect the final

$$\text{Cap}_{LF} + 1 \leq \text{Inspect}$$

$$\text{Cap}_{LB} + 1 \leq \text{Inspect}$$

$$\text{Cap}_{RF} + 1 \leq \text{Inspect}$$

$$\text{Cap}_{RB} + 1 \leq \text{Inspect}$$

$$5. \text{Inspect} + 3 \leq 30$$

2.2 Domain คือ

ช่วงเวลาที่ task แต่ละ  
 ชิ้นสามารถเริ่มได้ 30 นาที - 3 นาที  
 เวลาที่รวม  
 แล้ว

$$\text{Domain } S = \{1, 2, 3, \dots, 27\}$$

## 2. Consistency Techniques ตรวจสอบข้อ.สอดคล้อง

Backtracking มีข้อเสีย  $\rightarrow$  ตรวจสอบข้อ.ขัดแย้งซ้ำ จึงคิด Consistency เพื่อลด

Search space โดยตรวจสอบข้อ.ขัดแย้งก่อนลงล่างจริง, คัดค่าที่ขัดแย้งออกจากโดเมนตัวนั้น

### ประเภทของ Consistency

#### 1. Node Consistency (NC)

แนวคิด : ตัวแปร  $V$  จะ node-consistent ถ้าทุกค่าที่อยู่ใน  $\text{Domain}(V)$  ไม่ขัดกับ Unary

ถ้าพบค่าที่ละเมิด Unary  $\rightarrow$  ลบทิ้งจาก Domain ได้เลย

ตัวอย่าง : Crossword Puzzle

Variable ช่องดำในตาราง

Domain คำศัพท์ยาว 3, 4, 5 ตัวอักษร

Constraint ช่องยาว 3 ตัวเลือก Domain (3) เท่านั้น

ทำให้ Search Space ลดลงมาก

#### 2. Arc Consistency (AC)

แนวคิด : ตัวแปร  $x_i$  เป็น arc consistent กับ  $x_j$  ถ้าทุกค่าของ  $x_i$  มีอย่างน้อย 1 ค่าของ  $x_j$  ที่ทำให้ Constraint ระหว่าง  $x_i$  กับ  $x_j$  เป็นจริง

ตัวอย่าง : Domain  $A = \{3, 4, 5, 6, 7\}$

Domain  $B = \{1, 2, 3, 4, 5\}$

Constraint  $A < B$

คัดค่าที่ขัดแย้งออก เหลือ

Domain  $A = \{3, 4\}$

Domain  $B = \{4, 5\}$