

# Constraint Satisfaction Problems (CSPs)

---

## Định nghĩa tổng quát CSPs

Các bài toán thỏa mãn những ràng buộc, CSPs là đại diện cho một problem có tập hợp các ràng buộc được giải quyết bằng chính những phương pháp thỏa mãn những ràng buộc đó. Có độ phức tạp cao (đòi hỏi sự kết hợp giữa **heuristics** & **search method**).

Mỗi state của problem sẽ được chia thành các **variables**, mỗi biến cần được xác định miền giá trị (**domain**) & ràng buộc đi kèm với nó (**constraint**) hoặc giữa nó với các biến khác. Mỗi **variable** sẽ có 1 value nào đó thỏa được tất cả những ràng buộc của nó.

Để fomulate một CSP cần 3 thành phần chính cần phải xác định:

- **X**: tập hợp các **variables**.
- **D**: tập hợp các **domains** cho từng var. (Di là miền giá trị cho biến Xi)
- **C**: tập hợp các **constraints** mà state yêu cầu cho các var.

### Solution cho CSP: **consistent**

- Tất cả các X đều được gán 1 value cụ thể
- Các value đó đều thỏa mãn (không vi phạm) ràng buộc

### Example:

*Map coloring problem*: tô màu toàn bộ node mà không node nào kề cận cùng màu nhau.

*Task scheduling*: bài toán lập lịch, quản lí thời gian hợp lí & các ràng buộc thứ tự các công việc trước sau.

### Các loại ràng buộc:

*Unary constraint*: hạn chế giá trị của một biến duy nhất

*Binary constraint*: ràng buộc liên quan đến 2 biến

*Higher order constraint*: liên quan đến ba hoặc nhiều biến

*Global constraint* : liên quan đến một số biến tùy ý

## Constraint Propagation: Inference in CSPs

Lan truyền ràng buộc: khi xác định value cho một node thì sẽ giảm được constraint cho node khác.

### Node consistency:

Một single variable là **node consistency** nếu tất cả giá trị trong domain của nó là **unary constraint**.

### Arc consistency:

Một variable là **arc consistency** nếu tất cả giá trị trong domain của nó là **binary constraint**.

## Backtracking Search for CSPs

- Sử dụng tính chất giao hoán
- Thuật toán: tương tự như DFS
  - Chọn 1 value cho biến rồi tiếp tục đi (forward checking)
  - Cho đến khi value đó không hợp lệ thì quay lên bước  $k - 1$  (back checking)
- Example: mã đi tuần
- Các heuristic tối ưu: bằng cách điều khiển thứ tự các variables (MRV/DH) hay thứ tự các value cho 1 variable (LCV):
  - Minimum-remaining-values (MRV): chọn biến có giá trị hợp lệ ít nhất trước.
  - Degree heuristic (DH): chọn biến có nhiều ràng buộc nhất (degree cao nhất) thì giải quyết trước.
  - Least constraining value (LCV): chọn giá trị có ít ràng buộc với biến khác nhất (maximum flexibility).

## Local Search for CSPs

Min-conflicts heuristic: chọn biến có ít conflicts với các biến khác nhất

## The Structure of Problems

**slide 51**