

# LÝ THUYẾT TÍNH TOÁN

## BÀI 10: Định nghĩa giải thuật

---

Phạm Xuân Cường  
Khoa Công nghệ thông tin  
[cuongpx@tlu.edu.vn](mailto:cuongpx@tlu.edu.vn)

1. Khái niệm
2. Bài toán của Hilbert
3. Luận đề Church-Turing

## Khái niệm

---

- Một **giải thuật** là tập các lời chỉ dẫn đơn giản để thực hiện một vài nhiệm vụ nào đó
- Giải thuật = thủ tục = công thức
- Giải thuật đóng vai trò quan trọng cho rất nhiều nhiệm vụ khác nhau  
Ví dụ: tìm số nguyên tố, tìm ước số chung lớn nhất,...
- Trước thế kỉ XX, chưa tồn tại khái niệm giải thuật (**các khái niệm mang tính trực giác về giải thuật**)

# Bài toán của Hilbert

---

# Bài toán của Hilbert

- Năm 1900 nhà toán học David Hilbert đưa ra một bài toán có liên quan đến giải thuật
- Xét bài toán đa thức:
- Ví dụ:  $6x^3yz^2 + 3xy^2 - x^3 - 10$ 

Một nghiệm của đa thức là bộ các giá trị  $x, y, z$  sao cho đa thức có giá trị bằng 0

$$x = 5, y = 3, z = 0$$
- Bài toán của Hilbert là hãy đưa ra 1 giải thuật để kiểm tra 1 đa thức có nghiệm nguyên hay không

- Cần phải có định nghĩa về giải thuật mới có thể giải quyết được bài toán trên
- Alonzo Church và Alan Turing đã đưa ra một phép tính Lamda ( $\lambda$ ) để định nghĩa các giải thuật tương đương
- Mối liên hệ giữa khái niệm không hình thức của giải thuật và định nghĩa chính xác → **Luận đề Church-Turing**

# Luận đề Church-Turing

---



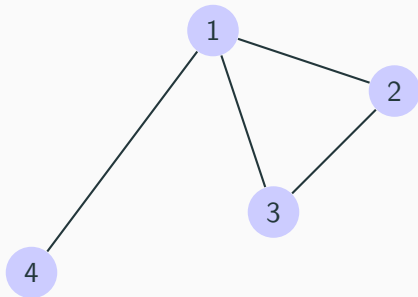
## Giải thuật $\equiv$ Máy Turing

Các cách để mô tả giải thuật:

- Đặc tả hình thức
  - Mô tả các trạng thái, bộ chữ, hàm chuyển dịch  $\rightarrow$  Là mô tả mức thấp nhất, đầy đủ nhất
- Đặc tả thực thi (Implementation-level specification) Sử dụng văn xuôi để mô tả
  - Nội dung của băng nhớ
  - Cách thức biểu diễn dữ liệu
  - Cách hoạt động của đầu đọc
- Đặc tả mức cao (High-level specification) Sử dụng văn xuôi để mô tả (Bỏ qua các chi tiết thực thi)
  - Mã giả (Pseudo-code)

## Ví dụ

Bài toán: Cho đồ thị  $G = (V, E)$  hãy cho biết đồ thị có liên thông hay không?



→ Ta đưa về bài toán đoán nhận ngôn ngữ:

$A = \{ \langle G \rangle \mid G \text{ là một đồ thị liên thông} \}$

→ Ta cần đưa ra một máy Turing đoán nhận  $A$

## Ví dụ (2)

- Biểu diễn đồ thị  $G$  thành một ngôn ngữ

Ví dụ:  $G = (\{1,2,3,4\}, \{(1,2), (1,3), (1,4), (2,3)\})$

$\rightarrow \langle G \rangle = (1,2,3,4)((1,2), (1,3), (1,4), (2,3))$

$\rightarrow \Sigma = \{ (, ), 1, 2, 3, 4, \dots \}$

## Ví dụ (3)

Mô tả ở mức cao của TM quyết định A

$M =$  "Trên dữ liệu vào  $\langle G \rangle$  là mã hóa của đồ thị  $G$ :

1. Chọn nút đầu tiên của đồ thị  $G$  và đánh dấu nó
2. Lặp lại giai đoạn sau đến khi không còn nút nào được đánh dấu
3. Với mỗi nút trong đồ thị  $G$ , đánh dấu nó nếu tồn tại một cạnh đi từ nó đến 1 nút đã đánh dấu
4. Kiểm tra tất cả các nút của  $G$  để xác định xem tất cả các nút đã được đánh dấu chưa
  - Nếu tất cả các nút đã được đánh dấu  $\rightarrow$  Chấp thuận
  - Nếu còn nút chưa được đánh dấu  $\rightarrow$  Bác bỏ

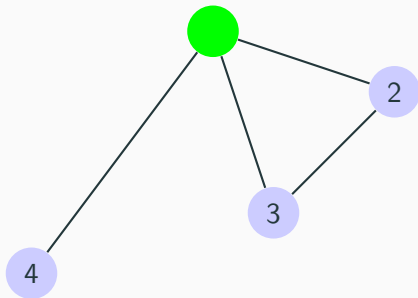
## Ví dụ (4)

Mô tả ở mức thực thi

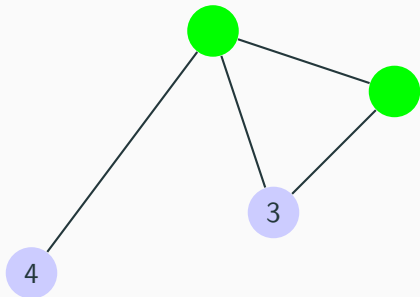
1. Kiểm tra chuỗi đầu vào đã đúng là mã hóa của 1 đồ thị chưa
  - Kiểm tra danh sách nút
    - Kiểm tra từng ký tự 1 xem nó có bị lặp lại không
  - Kiểm tra danh sách cạnh
    - ...
2. Đánh dấu nút đầu tiên → Thêm dấu chấm vào nút đầu tiên
3. Duyệt danh sách nút để tìm ra nút không được đánh dấu
  - ...

## Ví dụ (5)

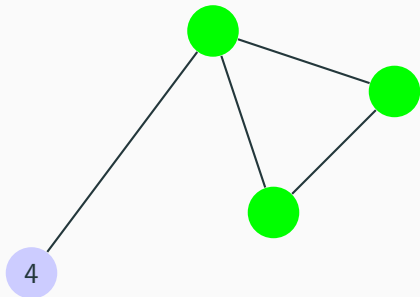
$$\langle G \rangle = (1,2,3,4)((1,2),(1,3),(1,4),(2,3))$$



## Ví dụ (6)

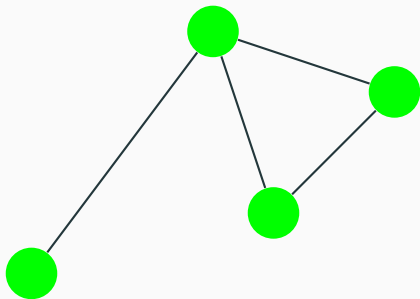


## Ví dụ (7)





## Ví dụ (8)



→ Đồ thị liên thông

Bài 1: Cho bộ chữ  $\Sigma = \{0,1\}$

- Hãy đưa ra biểu đồ trạng thái của NFA đoán nhận ngôn ngữ tương đương với biểu thức chính quy  $\Sigma^*0\Sigma$
- Mô tả định nghĩa hình thức của NFA trên
- Hãy đưa ra biểu đồ trạng thái của DFA tương đương với NFA trên và mô tả định nghĩa hình thức
- Hãy mô tả ngôn ngữ mà NFA trên đoán nhận

Bài 2: Cho ngôn ngữ  $L = \{a^n b^n c^m d^m \mid n, m > 0\}$ . Hãy chứng minh  $L$  là không phi ngữ cảnh.

Bài 3: Cho ngôn ngữ  $L = \{a^i b^j c^k \mid k = i + j; i, j, k \geq 0\}$

- Hãy đưa ra biểu đồ trạng thái cho máy Turing quyết định  $L$
- Hãy đưa ra lịch sử tính toán trên TM ở phần a với chuỗi aabbcccc

**Questions?**