ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Lý Thuyết NP-Đầy-Đủ

THUẬT TOÁN ỨNG DỤNG

Độ khó của bài toán

- Đánh giá độ khó của bài toán là ước lượng thời gian tính của thuật toán tốt nhất trong số tất cả các thuật toán giải bài toán kể cả các thuật toán đã biết lẫn các thuật toán còn chưa biết
- Hai cách tiếp cận:
  - Cách 1: tìm cách đưa ra đánh giá cận dưới độ phức tạp của bài toán (Dành cho khoá học Phân Tích Thuật Toán Nâng Cao)
  - Cách 2: tập trung vào việc chỉ ra mức độ khó của nó có thể so sánh với bất kỳ bài toán khó hiện biết (Lý thuyết NP-đầy-đủ)
- Việc đánh giá được độ phức tạp tính toán của bài toán giữ vai trò định hướng trong việc thiết kế thuật toán để giải bài toán đặt ra

Giới thiệu

- Từ những năm 1960, Steve Cook và Dick Karp đã quyết định rằng yêu cầu tối thiểu đối với một thuật toán hiệu quả là thời gian tính của nó phải là đa thức: $\mathcal{O}(n^c)$ với c là hằng số
- Người ta cũng nhận thấy rằng đối với nhiều lớp bài toán, việc tìm ra những thuật toán như vậy là rất khó khăn, hơn nữa chúng ta còn không biết là một thuật toán như vậy có tồn tại hay không
- Chính vì thế, Cook và Karp và một số người khác đã đưa ra định nghĩa lớp bài toán NP-đầy-đủ, mà cho đến hiện nay người ta vẫn tin rằng là không thể có thuật toán hiệu quả để giải chúng

Dick Karp (1972)

(Cook et. al., 1998)

Lớp bài toán P, NP

P là lớp các bài toán có thể giải được trong thời gian đa thức

- Bài toán về tính liên thông của đồ thị có thể giải được nhờ thuật toán với thời gian tính là $\mathcal{O}(n^2)$, vì vậy, nó là bài toán thuộc lớp P
- Bài toán nhân dãy ma trận giải được nhờ qui hoạch động với thời gian $\mathcal{O}(n^3)$, cũng thuộc vào lớp P

Lớp bài toán P, NP

P là lớp các bài toán có thể giải được trong thời gian đa thức

- Bài toán về tính liên thông của đồ thị có thể giải được nhờ thuật toán với thời gian tính là $\mathcal{O}(n^2)$, vì vậy, nó là bài toán thuộc lớp P
- Bài toán nhân dãy ma trận giải được nhờ qui hoạch động với thời gian $\mathcal{O}(n^3)$, cũng thuộc vào lớp P

NP là lớp các bài toán *kiểm chứng được trong thời gian đa thức*

nghĩa là đưa ra được thuật toán trong thời gian đa thức kiểm chứng tính chính xác của một bộ kết quả đầu ra với bộ dữ liệu đầu vào tương ứng

- Bài toán kiểm tra tính hợp số: "Có phải số n là hợp số?"
- Bài toán tìm chu trình Hamilton, việc kiểm tra dãy đỉnh $v_1, v_2, \ldots, v_n, v_1$ có là chu trình Hamilton của đồ thị đã cho hay không có thể thực hiện sau thời gian đa thức

Lớp bài toán P, NP, NPC

NP-đầy-đủ là lớp các bài toán trong lớp NP và khó không kém bất cứ bài toán nào trong NP

Đây là lớp bài toán khó nhất trong lớp NP

Lớp bài toán P, NP, NPC

NP-đầy-đủ là lớp các bài toán trong lớp NP và khó không kém bất cứ bài toán nào trong NP

Đây là lớp bài toán khó nhất trong lớp NP

Vì sao không nói "Giải được trong thời gian hàm mũ" hay "Giải được không trong thời gian đa thức"?

$\mathcal{O}(n^{100})$ và $\mathcal{O}(2^n)$

- Thuật toán $\mathcal{O}(n^{100})$ vẫn là thuật toán đa thức, tuy nhiên thời gian tính là không có tính ứng dụng thực tế
- Đa phần các thuật toán đa thức có thời gian tính nhỏ hơn rất nhiều
- Khi một thuật toán đa thức được tìm ra, nhiều thuật toán hiệu quả mới từ đó sẽ được khám phá

Mối quan hệ giữa P, NP, NPC

- $P \subseteq NP$ (chắc chắn)
- $NPC \subseteq NP$ (chắc chắn)
- $P = NP$ (hoặc $P \subset NP$, hoặc $P \neq NP$) ???
- $NPC = NP$ (hoặc $NPC \subset NP$, hoặc $NPC \neq NP$) ???

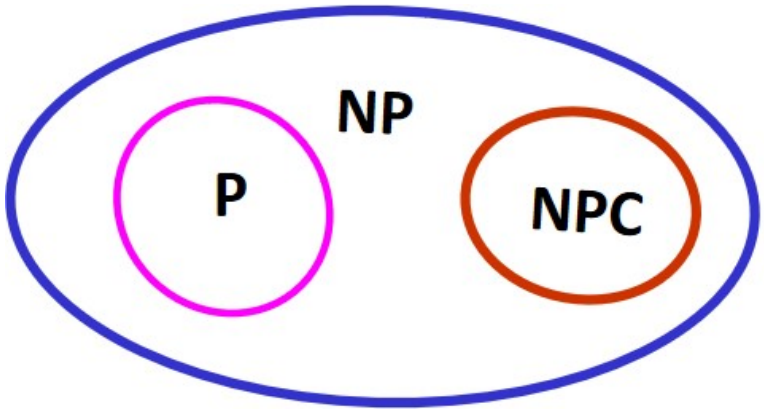
Mối quan hệ giữa P, NP, NPC

- $P \subseteq NP$ (chắc chắn)
- $NPC \subseteq NP$ (chắc chắn)
- $P = NP$ (hoặc $P \subset NP$, hoặc $P \neq NP$) ???
- $NPC = NP$ (hoặc $NPC \subset NP$, hoặc $NPC \neq NP$) ???

$P = NP$?

- Một trong những vấn đề hóc búa nhất và là trung tâm của lý thuyết tính toán đó là chứng minh hay bác bỏ đẳng thức này!
- Cho đến hiện nay vấn đề này vẫn còn là vấn đề mở

Mối quan hệ giữa P, NP, NPC



- Quan điểm của hầu hết các nhà nghiên cứu khoa học máy tính là:

$$P \subset NP, NPC \subset NP, P \cap NPC = \emptyset$$

Vì sao cần quan tâm đến lý thuyết NP-đầy-đủ?

- Nếu một bài toán được chứng minh là thuộc lớp NP-đầy đủ thì ta có được bằng chứng về độ khó của bài toán
- Không lãng phí thời gian để cố gắng tìm ra thuật toán hiệu quả cho bài toán NP-đầy-đủ
- Thay vào đó, hãy tập trung vào thiết kế thuật toán gần đúng hoặc heuristic/metaheuristic hoặc tìm lời giải hiệu quả cho một số trường hợp đặc biệt của bài toán
- Một số bài toán nhìn bề ngoài thì rất dễ, nhưng thực tế là khó (thuộc lớp NP-đầy-đủ)

Bài toán quyết định và Bài toán tối ưu

Bài toán tối ưu là bài toán yêu cầu tìm ra lời giải tối ưu max hoặc min

Ví dụ bài toán tối ưu: `SHORTEST-PATH`

- Cho G, u, v, hãy tìm một đường đi từ u đến v qua ít cạnh nhất

Bài toán quyết định và Bài toán tối ưu

Bài toán tối ưu là bài toán yêu cầu tìm ra lời giải tối ưu max hoặc min

Ví dụ bài toán tối ưu: `SHORTEST-PATH`

- Cho G, u, v, hãy tìm một đường đi từ u đến v qua ít cạnh nhất

Bài toán quyết định là bài toán mà đầu ra chỉ có thể là 'YES' hoặc 'NO' (đúng/sai, 1/0, chấp nhận/từ chối)

- Đối với một bài toán quyết định, có những bộ dữ liệu vào của nó có câu trả lời (đầu ra) là 'YES' và cũng có những bộ dữ liệu vào có câu trả lời là 'NO'
- Những bộ dữ liệu vào với câu trả lời 'YES' ('NO') sẽ được gọi là bộ dữ liệu vào 'YES' ('NO')

Ví dụ bài toán quyết định: `PATH`

- Cho G, u, v, k, hỏi có tồn tại hay không một đường đi từ u đến v qua tối đa k cạnh?

Bài toán quyết định và Bài toán tối ưu

Lớp NP-đầy-đủ chỉ gồm các bài toán quyết định!

- Đối với bài toán tối ưu, nếu kiểm chứng được tính tối ưu và chính xác của bộ kết quả đầu ra trong thời gian đa thức thì cũng có thể tìm được lời giải tối ưu trong thời gian đa thức
- Việc qui dẫn giữa các bài toán quyết định dễ dàng hơn so với giữa các bài toán tối ưu

Dạng quyết định của bài toán tối ưu

- Xét bài toán tối ưu hoá:

(PO): $\max\{f(x) : x \in D\}$

- Bài toán dạng quyết định (PD) tương ứng với bài toán tối ưu (PO) là:

(PD): "Cho giá trị k, hỏi có tìm được $u \in D$ sao cho $f(u) \geq k$?"

- Lời giải của bài toán tối ưu dẫn ra trực tiếp lời giải cho bài toán quyết định tương ứng
- Nếu bài toán quyết định tương ứng với một bài toán tối ưu có thể giải được hiệu quả (chẳng hạn, bằng thuật toán đa thức) thì bài toán tối ưu đó cũng giải được hiệu quả (bằng thuật toán đa thức)

Mối liên hệ giữa (PO) và (PD)

- Giả thiết rằng hàm f nhận giá trị nguyên trên D và ta biết α_0 và β_0 là cận dưới và cận trên của giá trị hàm f trên D
- Giả sử A là thuật toán giải bài toán (PD) với độ phức tạp $\mathcal{O}(n^c)$

Bước lặp $k = 0, 1, 2, \ldots$

- Tính $\gamma_k = (\alpha_k + \beta_k)/2$;
- Nếu thuật toán A tìm được $x_k \in D$ thoả mãn $f(x_k) > \gamma_k$ thì $\alpha_{k+1} = \gamma_k, \beta_{k+1} = \beta_k$;
- Trái lại, đặt $\alpha_{k+1} = \alpha_k, \beta_{k+1} = \gamma_k$;
- Chuyển sang bước $k + 1$;

Rõ ràng sơ đồ trên cho thời gian tính toán $\mathcal{O}(\log(\beta_0 - \alpha_0)n^c)$ để giải bài toán (PO)

(PO) và (PD): Bài toán tìm tập độc lập cực đại

MIS:

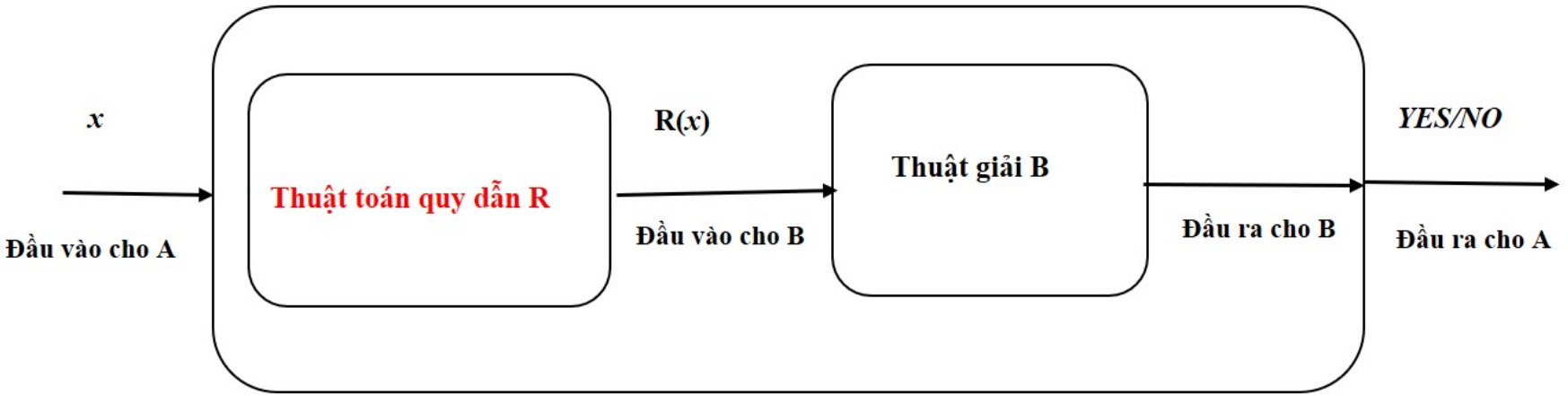
- Cho đồ thị vô hướng $G = (V, E)$. Một tập con các đỉnh của đồ thị mà hai đỉnh bất kỳ trong nó là không kề nhau trên đồ thị được gọi là tập độc lập của đồ thị
- Yêu cầu: Tìm tập độc lập với lực lượng lớn nhất (tập độc lập cực đại)

- Bài toán dạng quyết định tương ứng có dạng: "Cho số nguyên dương k, hỏi đồ thị có chứa tập độc lập với lực lượng ít ra là k hay không?"
- Do $1 \leq k_{\max} \leq n$, ($k_{\max}$ là lực lượng của tập độc lập cực đại), nên nếu bài toán dạng quyết định giải được bằng thuật toán đa thức thì bài toán tối ưu cũng giải được bằng sơ đồ trình bày trên sau không quá $\lceil \log n \rceil$ lần áp dụng thuật toán giải bài toán dạng quyết định

Phép qui dẫn trong thời gian đa thức

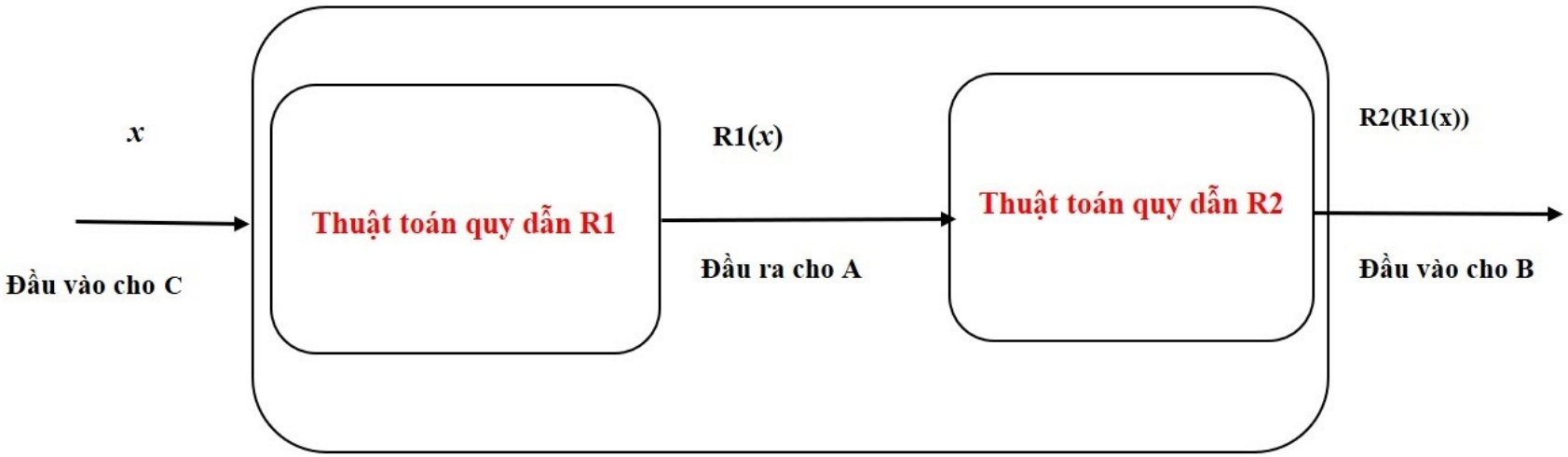
- Giả sử A và B là hai bài toán quyết định. Ta nói bài toán A có thể qui dẫn trong thời gian đa thức về bài toán B nếu tồn tại thuật toán thời gian đa thức R cho phép biến đổi bộ dữ liệu vào x của A thành bộ dữ liệu vào $R(x)$ của B sao cho x là bộ dữ liệu 'YES' (nghĩa là bộ dữ liệu mà câu trả lời cho nó là 'YES') của A khi và chỉ $R(x)$ là bộ dữ liệu 'YES' của B
- Trong phần tiếp theo ta chỉ xét phép qui dẫn sau thời gian đa thức, vì thế để ngắn gọn, ta sẽ gọi là phép qui dẫn thay cho phép qui dẫn sau thời gian đa thức

Phép qui dẫn trong thời gian đa thức



- Kí hiệu A qui dẫn về B: $A \prec B$
- Nếu tồn tại thuật toán đa thức giải B thì A cũng sẽ được giải trong thời gian đa thức
- A không khó hơn B (hay B không dễ hơn A)
- Nếu bài toán A là khó (ví dụ thuộc lớp NP-đầy-đủ), thì bài toán B cũng khó
- Phép qui dẫn này được sử dụng để chỉ ra một bài toán là khó.

Phép qui dẫn



- Nếu C $\prec$ A và A $\prec$ B, thì C $\prec$ B

NP-đầy-đủ và NP-khó

Định nghĩa

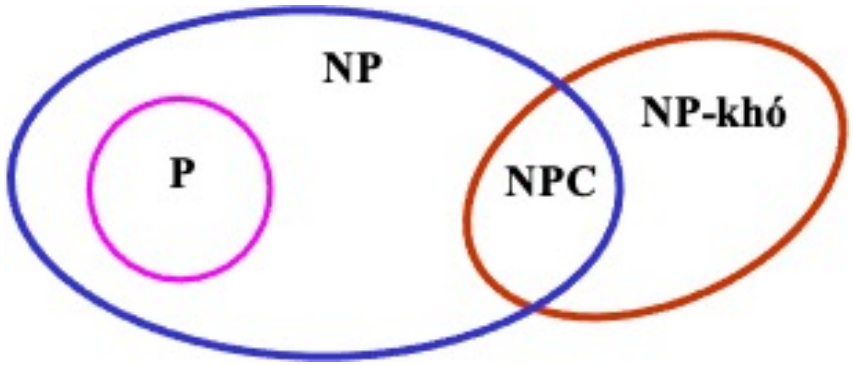
Bài toán quyết định A được gọi là NP-đầy đủ nếu như

1. A là bài toán trong NP
2. Mọi bài toán trong NP đều có thể qui dẫn về A

Nếu bỏ đi điều kiện (1) mà chỉ thỏa mãn điều kiện (2) thì A được gọi là NP-khó

- Như vậy, có thể nói khái niệm về "bài toán khó nhất" trong lớp NP được xây dựng trên cơ sở phép qui dẫn
- Nếu tất cả các bài toán trong NP có thể qui dẫn về một bài toán A thì A khó không kém bất cứ bài toán nào trong số chúng
- Chỉ với điều kiện (2), nếu tồn tại thuật toán đa thức để giải A thì sẽ kéo theo sự tồn tại thuật toán đa thức để giải mọi bài toán trong NP

NP-đầy-đủ và NP-khó



Hình: Bức tranh tạm thời chi tiết hơn

Khó khăn lớn nhất là việc tìm ra được một bài toán B thuộc NP để qui dẫn về A. Bởi vì hễ chúng ta đã có một bài toán NP-đầy-đủ thì có thể chứng minh nhiều bài toán khác là NP-đầy-đủ nhờ sử dụng kết quả sau đây:

Bổ đề

Nếu B $\in$ NP-đầy-đủ, A $\in$ NP, và B $\prec$ B, khi đó A $\in$ NP-đầy-đủ

NP-đầy-đủ và NP-khó

Vậy làm thế nào để chứng minh bài toán NP-đầy-đủ đầu tiên?

NP-đầy-đủ và NP-khó

Vậy làm thế nào để chứng minh bài toán NP-đầy-đủ đầu tiên?

- Bài toán về tính thực tiễn của mạch logic `CIRCUIT-SAT` (Circuit-satisfiability) là bài toán đầu tiên được chứng minh là bài toán NP-đầy-đủ bằng cách đưa ra một thuật toán qui dẫn đa thức từ bất kì một bài toán L nào thuộc lớp NP
- Tuy nhiên về mặt lịch sử `CIRCUIT-SAT` không phải là bài toán NP-đầy-đủ đầu tiên, mà đó là bài toán `SAT`

Bài toán về tính thực tiễn của mạch logic

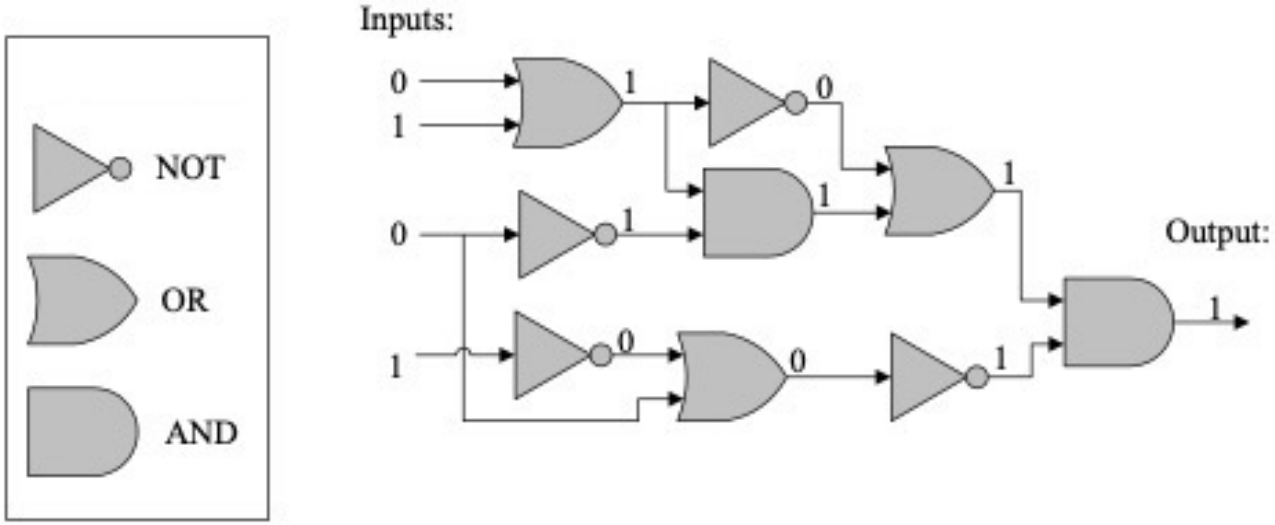
CIRCUIT-SAT: Cho một mạch lôgic với đầu vào gồm n giá trị, hỏi có tồn tại một đầu vào của mạch để đầu ra của mạch là TRUE, hay mạch luôn đưa ra FALSE?

Hình: Mạch không bật được với mọi $x_1 \in \{0, 1\}$

Hình: Mạch nhận giá trị TRUE với $x_1, x_2 = 01, 10$, hoặc 11, như vậy câu trả lời là 'YES'

Định lí: CIRCUIT-SAT là NP-đầy-đủ (Cook, 1971)

1. CIRCUIT-SAT $\in$ NP: Cho 1 đầu vào, dễ dàng tính được đầu ra tương ứng sau thời gian đa thức nhờ sử dụng thuật toán duyệt đồ thị



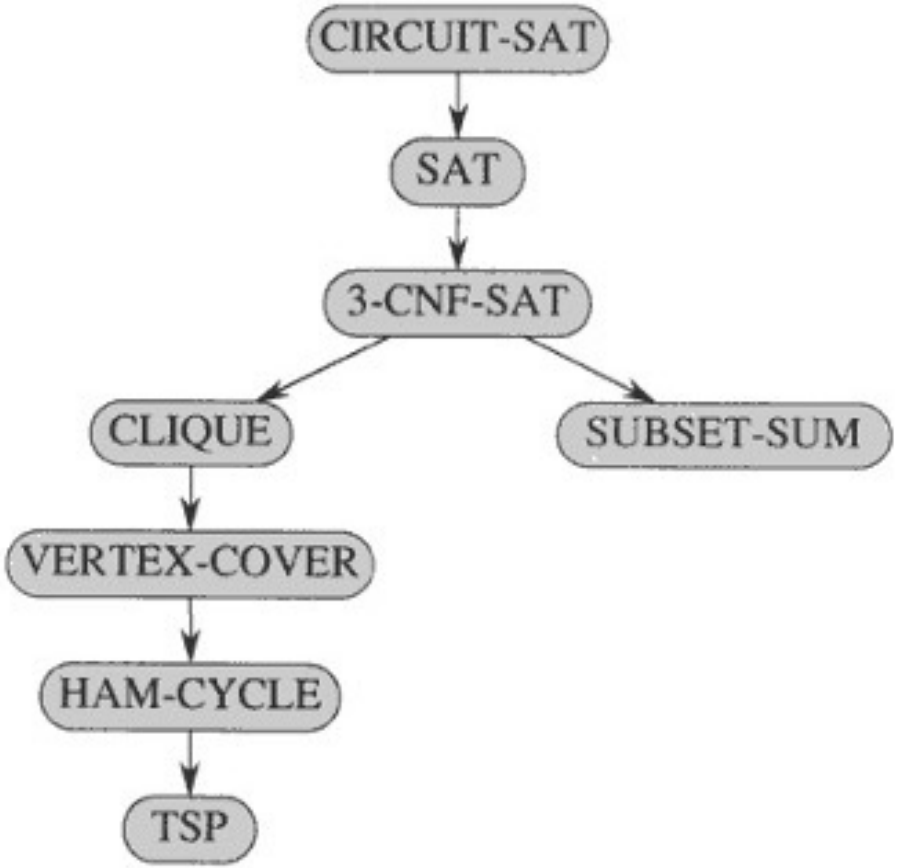
CIRCUIT-SAT là NP-đầy-đủ

2. Việc chứng minh mọi bài toán trong NP đều qui dẫn được về CIRCUIT-SAT là khá phức tạp. Ý tưởng chứng minh được phác thảo như sau:
   - Mọi bài toán trong NP đều có thể tính được nhờ mạch lôgic
   - Mạch lôgic này có số thành phần giới hạn bởi đa thức và vì thế cũng tính được sau thời gian đa thức

Sơ đồ chứng minh L là NP-đầy-đủ

1. Chứng minh L $\in$ NP: thường là dễ
2. Chọn bài toán L' là NP-đầy-đủ (hoặc NP-khó)
3. Xây dựng thuật toán thời gian tính đa thức xác định hàm f thực hiện việc biến đổi bộ dữ liệu của L' thành bộ dữ liệu của L
4. Chứng minh: x là bộ dữ liệu 'YES' của L' khi và chỉ khi $f(x)$ là bộ dữ liệu 'YES' của L

Cấu trúc cây qui dẫn các bài toán NP-đầy-đủ cơ bản



Một số bài toán trên cây qui dẫn NP-đầy-đủ

CIRCUIT-SAT

- Cho một mạch lôgic với đầu vào gồm n giá trị
- Hỏi có tồn tại một đầu vào của mạch để đầu ra của mạch là TRUE, hay mạch luôn đưa ra FALSE?
- $L \prec$ CIRCUIT-SAT, $\forall L \in NP$

SAT

- Cho một biểu thức lôgic tạo thành bởi n biến Bun: $x_1, x_2, \ldots, x_n$, các mệnh đề, các toán tử lôgic AND $\wedge$, OR $\vee$, NOT $\neg$, ... và các dấu ngoặc '(', ')'
- Hỏi có tồn tại một bộ giá trị của các biến Bun để cho biểu thức nhận giá trị TRUE?
- CIRCUIT-SAT $\prec$ SAT

Một số bài toán trên cây qui dẫn NP-đầy-đủ

3-CNF-SAT

- Cho một biểu thức lôgic dưới dạng 3-CNF (Conjunctive Normal Form), nghĩa là biểu thức Bun cấu thành từ hội của các mệnh đề mà mỗi mệnh đề là tuyển của đúng 3 toán hạng, mỗi toán hạng là 1 biến Bun (x) hoặc phủ định của nó ($\neg x$)
- Hỏi có tồn tại một bộ giá trị của các biến số để cho biểu thức nhận giá trị TRUE?
- SAT $\prec$ 3-CNF-SAT

SUBSET-SUM

- Cho tập S gồm n số nguyên dương $a_1, \ldots, a_n$ và số nguyên dương t
- Hỏi có thể tìm được tập con S' của S với tổng các số trong S' là bằng t?
- 3-CNF-SAT $\prec$ SUBSET-SUM

Một số bài toán trên cây qui dẫn NP-đầy-đủ

CLIQUE

- Một bè của đồ thị vô hướng $G = (V, E)$ là một tập con các đỉnh $S \subseteq V$ sao cho mỗi cặp đỉnh trong đó được nối bởi một cạnh $\in E$
- Nói một cách khác, một bè là một đồ thị con đầy đủ của G
- Lực lượng của một bè là số lượng đỉnh của bè đó
- Bài toán tối ưu: Tìm bè có lực lượng cực đại
- Bài toán quyết định: Hỏi có tồn tại bè lực lượng k cho trước trong G hay không?
- 3-CNF-SAT $\prec$ CLIQUE

Một số bài toán trên cây qui dẫn NP-đầy-đủ

VERTEX-COVER

- Một phủ đỉnh của đồ thị vô hướng $G = (V, E)$ là một tập con các đỉnh của đồ thị $S \subseteq V$ sao cho mỗi cạnh của đồ thị có ít nhất một đầu mút trong S
- Lực lượng của một phủ đỉnh là số lượng đỉnh của phủ đỉnh đó
- Bài toán tối ưu: Tìm phủ đỉnh có lực lượng cực tiểu
- Bài toán quyết định: Hỏi có tồn tại phủ đỉnh lực lượng k cho trước trong G hay không?
- CLIQUE $\prec$ VERTEX-COVER

Một số bài toán trên cây qui dẫn NP-đầy-đủ

HAM-CYCLE

- Hỏi đồ thị vô hướng $G = (V, E)$ có chứa chu trình Hamilton không?
- VERTEX-COVER $\prec$ HAM-CYCLE

TSP

- Cho ma trận chi phí $C = [C_{ij}$ giữa các thành phố và một số nguyên k:
- Bài toán tối ưu: Tìm hành trình người du lịch với tổng chi phí nhỏ nhất
- Bài toán quyết định: Hỏi có tồn tại một hành trình của người du lịch với tổng chi phí không vượt quá số k hay không?
- HAM-CYCLE $\prec$ TSP

Định lí: TSP là NP-đầy-đủ

Chứng minh:

1. TSP $\in$ NP: việc kiểm tra xem dãy các thành phố đã cho có phải là hành trình hợp lệ với chi phí không vượt quá k có thể dễ dàng thực hiện xong trong thời gian đa thức
2. Chứng minh TSP là NP-khó bằng phép qui dẫn HAM-CYCLE $\prec$ TSP

Qui dẫn HAM-CYCLE $\prec$ TSP

- Với mỗi trường hợp đồ thị $G = (V, E)$ của HAM-CYCLE, ta xây dựng một trường hợp tương ứng của TSP $< G', C, 0 >$ trong thời gian đa thức như sau:
  - $G' = (V, E')$, với $E' = \{< i, j >: i, j \in V, i \neq j\}$ và
  - hàm chi phí C được định nghĩa như sau:
    - $C(i, j) = 0$ nếu $(i, j) \in E$,
    - $C(i, j) = 1$, nếu trái lại
- Nếu G có một chu trình Hamilton H, thì H cũng là một hành trình hợp lệ trên G' với chi phí tối đa là 0
- Nếu G' có một hành trình H' với chi phí tối đa là 0, thì mỗi cạnh trên H' đều có chi phí bằng 0, vì vậy các cạnh đó đều $\in E$, và H' cũng là một chu trình Hamilton trên G

Các hướng tiếp cận giải bài toán NP-khó

Còn nhớ 4 mô hình giải bài căn bản ?

Các hướng tiếp cận giải bài toán NP-khó

Còn nhớ 4 mô hình giải bài căn bản ?

Duyệt nhánh cận (Lecture 3)

- Thời gian chạy lâu
- Thường chỉ đưa ra được đáp án tối ưu trong thời gian chấp nhận được với kích thước đầu vào đủ nhỏ hoặc với một số trường hợp đặc thù
- Khó ước lượng chính xác độ phức tạp tính toán

Các hướng tiếp cận giải bài toán NP-khó

Chia để trị / Qui hoạch động (Lecture 4/5)

- Có thể đưa ra được đáp án tối ưu trong mốt số bài toán (ví dụ: `TSP`, `KNAPSAC`)
- Độ phức tạp tính toán vẫn là hàm mũ
- Có thể sử dụng như là một phần của thuật toán heuristic để đưa ra được một lời giải chấp nhận được

Các hướng tiếp cận giải bài toán NP-khó

Thuật toán tham lam (Lecture 9)

- Nhanh chóng đưa ra được một lời giải chấp nhận được (feasible solution)
- Hay được dùng để tạo một lời giải khởi đầu cho các thuật toán heuristic/metaheuristic phức tạp
- Tính ứng dụng vào các bài toán thực tế cao, nhất là các bài toán đòi hỏi thời gian thực (real-time) với độ tốt của lời giải là chấp nhận được
- Về khoa học lý thuyết, thường được sử dụng trong lĩnh vực thuật toán xấp xỉ

Các hướng tiếp cận giải bài toán NP-khó

Thuật toán xấp xỉ (Approximation Algorithms)

- Thường là các thuật toán tham lam hay heuristic đơn giản
- Luôn chứng minh được chắc chắn độ tốt của lời giải của thuật toán đề xuất so với lời giải tối ưu luôn nằm trong giới hạn cận tỉ lệ ρ xác định
- Ít có tính ứng dụng trong các bài toán thực tế
- Là một lĩnh vực lý thuyết khó, thường xuất hiện trong các khóa học thạc sĩ chuyên ngành Khoa học máy tính

Các hướng tiếp cận giải bài toán NP-khó

Thuật toán heuristic

- Là các kỹ thuật giải bài được phát triển *dựa trên kinh nghiệm* phân tích các đặc điểm của bài toán để đưa ra một hướng tiếp cận cho lời giải chấp nhận được đủ tốt (không đảm bảo là tối ưu)
- Hay sử dụng các kỹ thuật tìm kiếm địa phương (local search) để cải tiến lời giải hiện biết
- Có thể cùng một lúc áp dụng nhiều thuật toán heuristic khác nhau, gọi là multiheuristic
- Là nền tảng cho việc phát triển các thuật toán metaheuristic

Các hướng tiếp cận giải bài toán NP-khó

Thuật toán metaheuristic

- Là các mô hình phát triển thuật toán cấp cao dùng để đưa ra các chiến lược điều khiển và điều chỉnh các thuật toán heuristic bằng cách thay đổi linh hoạt các tham số trong mô hình
- Lời giải bài toán thường sẽ được cải tiến so với các lời giải thuật toán heuristic đơn giản khi chọn được các bộ tham số của mô hình đủ tốt
- Thường phải chạy thí nghiệm nhiều lần để tìm ra được bộ tham số thích hợp
- Thời gian chạy và bộ nhớ sử dụng thường rất lớn và ít khả năng ứng dụng vào các bài toán thực tế đòi hỏi xử lý thời gian thực hoặc dữ liệu lớn
- Một số mô hình điển hình: Thuật toán di truyền/tiến hóa (genetic/evolutionary algorithms), tìm kiếm tabu (tabu search), mô phỏng tôi luyện thép (simulated annealing), tìm kiếm vùng lân cận linh hoạt (variable neighborhood search), tìm kiếm vùng lân cận không gian lớn (large neighborhood search), thuật toán đàn kiến (ant colony optimization), ...

Các hướng tiếp cận giải bài toán NP-khó

Học tăng cường (Reinforcement Learning)

- Là một trong ba nhánh chính của học máy, có thể áp dụng cho các bài toán tối ưu mô hình hoá được dưới dạng một quá trình ra quyết định tuần tự
- Xây dựng một môi trường mô phỏng cho phép tác nhân (agent) tương tác với môi trường đó thông qua việc thực hiện một chuỗi các hành động (action) để tìm lời giải. Một hướng học tăng cường phổ biến đó là phương pháp học dựa trên giá trị (value-based learning):
  - Tác nhân rút kinh nghiệm và ra quyết định từng bước dựa trên một hàm $Q(s, a)$ ước lượng độ tốt của hành động a đối với trạng thái s
  - Trong trường hợp không gian trạng thái đủ nhỏ, ta có thể biểu diễn dưới dạng bảng giá trị tương ứng giữa s, a và hàm $Q(s, a)$, được cập nhật liên tục dựa trên kinh nghiệm mà tác nhân đã trải qua
  - Trong trường không gian trạng thái rất lớn, ta có thể áp dụng phương pháp học tăng cường sâu (DRL), dùng mạng nơron DNN để biểu diễn mối quan hệ giữa s, a và $Q(s, a)$
- Bên cạnh các phương pháp truyền thống ở trên, đây là một hướng tiếp cận thời sự và hiện đại

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict