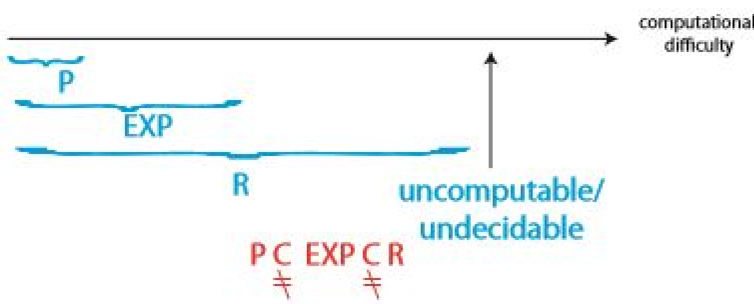
강의 23: 계산 복잡도

강의 개요

- P, EXP, R
- 대부분의 문제들은 계산 불가능하다
- NP
- 난해성 & 완전성
- 리덕션

정의:

- P = {다항 (n^c) 시간에 풀 수 있는 문제들}
  (이 강의에서 다루는 모든 것들)
- EXP = {지수 (2^{n^c}) 시간에 풀 수 있는 문제들}
- R = {유한 시간에 풀 수 있는 문제들} "재귀적" [Turing 1936; Church 1941]



예시

- 음수 가중치 사이클 탐지 $\in$ P
- $n \times n$ 체스 $\in$ EXP 그러나 $\notin$ P
  주어진 보드 배열에서 누가 이기는가?
- 테트리스 $\in$ EXP 그러나 $\in$ P 인지는 모름
  주어진 보드와 조각들에서 살아남기

정지 문제:

컴퓨터 프로그램이 주어졌을 때, 그것이 한 번이라도 정지하는가?

- 계산 불가능 ($\notin$ R):이것을 푸는 알고리즘은 없다 (모든 입력값에 대해 유한 시간에 맞게 푸는 것)
- 결정 문제: 답은 **예** 또는 **아니오**

대부분의 결정 문제들은 계산 불가능

- 프로그램 ≈ 이진 문자열 ≈ 음수가 아닌 정수 $\in N$
- 결정 문제 = 이진 문자열 (≈ 음수가 아닌 정수)에서 {예 (1), 아니오 (0)}로 변환하는 함수
- ≈ 비트의 유한한 나열 ≈ 실수 ∈R
  |N| << |R|:고유한 음수가 아닌 정수를 실수에 할당할 수 없음 (R 셀 수 없음)
- ⇒ 모든 문제에 대한 프로그램이 없음
- 모든 프로그램은 한 개의 문제를 푼다
- ⇒ 대부분의 문제들은 풀 수 없음

NP

NP ={결정 문제는 "행운" 알고리즘을 통해 다항 시간에 해결 가능}. "행운" 알고리즘은 모든 경우의 수를 시도할 필요 없이 항상 운 좋게 옳은 추측을 함

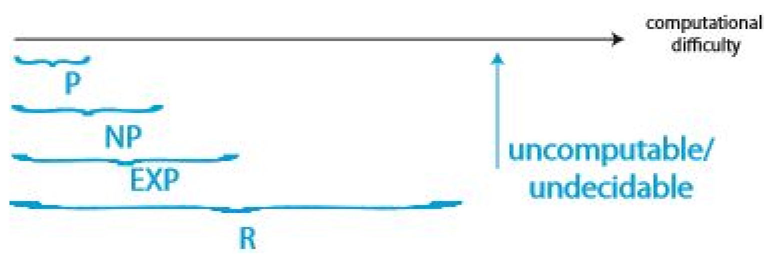
- 비결정적 모델: 알고리즘은 추측을 하고 예 또는 아니오를 도출
- 만약 가능하다면 추측이 예를 도출하는 것이 보장됨 (불가능하다면 아니오)

다른 말로, NP ={다항 시간에 "확인"될 수 있는 해답이 있는 결정 문제들}. 답 = YES이면, "증명" 가능 & 다항 시간 알고리즘으로 확인 가능

예시

테트리스 ∈ NP

- 비결정적 알고리즘: 각 움직임들을 추측하자. 살아남았는가?
- **예**에 대한 증명: 움직임들의 리스트 (테트리스의 규칙은 쉬움)



P $\neq$ NP

큰 추측 ($1,000,000 가치)

- ≈ 운을 설계할 수 없다
- ≈ 해답을 만드는 것은 그것을 확인하는 것보다 어렵다

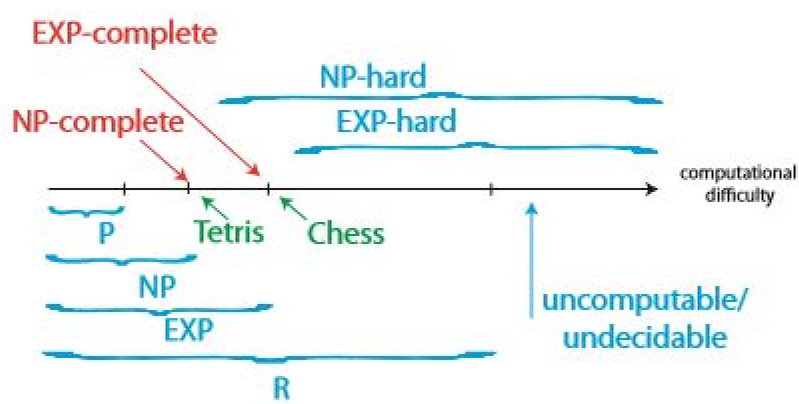
난해성과 완전성

주장:

만약 P ≠ NP이면, 테트리스 ∈ NP - P 이다.
[Breukelaar, Demaine, Hohenberger, Hoogeboom, Kosters, Liben-Nowell 2004]

이유:

테트리스는 NP-난해 = 모든 문제 "만큼 어렵다" ∈ NP. NP-완전 = NP ∩ NP-난해



유사한 경우

체스는 EXP-완전 = EXP ∩ EXP-난해. EXP-난해는 EXP의 모든 문제만큼 어렵다. 만약 NP $\neq$ EXP이면, 체스는 $\notin$ EXP \ NP. NP EXP $\neq$ 인지 아닌지 또한 미해결 문제이지만 덜 유명하고 덜 “중요하다”.

리덕션

문제를 이미 해결법을 아는 문제로 바꾼다 (처음부터 해결하는 대신)

- 가장 보편적인 알고리즘 디자인 기술
- 비가중치 최소 경로 → 가중치 (가중치 = 1로 둔다)
- 최소 곱 경로 → 최단 경로 (로그를 취한다) [PS6-1]
- 최장 경로 → 최단 경로 (가중치를 음수로) [Quiz 2, P1k]
- 최단 순서 순회 → 최단 경로 (그래프의 k개의 복사본) [Quiz 2, P5]
- 최소 가격 구멍난 탱크 경로 → 최단 경로 (그래프 리덕션) [Quiz 2, P6]

위의 모든 문제들은 원-콜 리덕션: 문제 A → 문제 B → 해답 B → 해답 A
멀티콜 리덕션: B를 제한 없이 호출해서 A를 해결 — 이런 의미에서, 모든 알고리즘은 문제를 축소시킨다 → 계산 모델
NP-완전 문제들은 다항 시간 리덕션을 사용해서 서로간에 축소가 가능하다 (같은 난이도). 따라서 NP-난해성을 증명하는 데에 리덕션을 사용할 수 있다 — 예를 들어 3-분할 → 테트리스

NP-완전 문제 예시

- 배낭 문제 (유사 다항, 다항 아님)
- 3-분할 문제: n개의 정수가 주어졌을 때, 같은 합을 가지는 세 개의 숫자들의 그룹으로 나눌 수 있는가?
- 외판원 문제: 그래프의 모든 정점을 방문하는 최단 경로
  — 결정 버전: 최소 가중치 $\leq x$ 인가?
- k개의 문자열의 최장 공통 부분 문자열
- 지뢰 찾기, 스도쿠, 대부분의 퍼즐
- SAT: 주어진 논리식(그리고, 또는, not)이 참이 될 수 있는가? x 그리고 not x → 아니오

- 3차원에서의 최단 경로
- 주어진 그래프의 3색 색칠
- 주어진 그래프에서 가장 큰 클릭 찾기