

SAT Problem

14004 구재현

what's SAT?

- $\vee \Rightarrow$ OR, $\wedge \Rightarrow$ AND
- 1-SAT : $a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge a_5 (\dots)$
- 2-SAT : $(\sim a_1 \vee a_2) \wedge (\sim a_3 \vee a_4) \wedge (a_1 \vee \sim a_4) (\dots)$
- 3-SAT : $(a_1 \vee \sim a_2 \vee a_3) \wedge (a_3 \vee \sim a_4 \vee a_6) \wedge (\sim a_1 \vee a_4 \vee a_5) (\dots)$
- more coming on...

Practice (1-SAT)

- $a1 \wedge \sim a2 \wedge a3 \wedge \sim a4 \wedge a5 (\dots)$
- $a1 = \sim a2 = a3 = \sim a4 (\dots) = \text{TRUE}$ 여야지 만족. 참 쉽죠?
- 해당 논리절을 참으로 만드는 값을 찾는 것을 SAT 문제를 푼다 라고 함
- 1-SAT은 자명하게도, 다항 시간 (선형 시간) 에 풀림.
- 다른 경우는?

why 3-SAT?

- SAT 문제 중에서 유독 3-SAT 문제가 유명한 이유는
- k -SAT ($k > 3$) 문제를 모두 3-SAT으로 치환할 수 있기 때문!
- $(a_1 \vee a_2 \vee a_3 \vee a_4 \vee a_5)^{\wedge}(\dots)$ 일 경우
- $(a_1 \vee a_2 \vee s_1)^{\wedge}(\sim s_1 \vee a_3 \vee s_2)^{\wedge}(\sim s_2 \vee a_4 \vee a_5) \rightarrow 3\text{-SAT}$
- 정확한지는 모름... 대략 이런식이라고 카더라!
- 때문에, 사실상 2-SAT과 3-SAT만이 의미를 가짐

NP-Complete

MY HOBBY:
EMBEDDING NP-COMPLETE PROBLEMS IN RESTAURANT ORDERS

CHOTCHKIES RESTAURANT	
~ APPETIZERS ~	
MIXED FRUIT	2.15
FRENCH FRIES	2.75
SIDE SALAD	3.35
HOT WINGS	3.55
MOZZARELLA STICKS	4.20
SAMPLER PLATE	5.80
~ SANDWICHES ~	
BARBECUE	6.55

WE'D LIKE EXACTLY \$15.05
WORTH OF APPETIZERS, PLEASE.

... EXACTLY? UHH...

HERE, THESE PAPERS ON THE KNAPSACK
PROBLEM MIGHT HELP YOU OUT.

LISTEN, I HAVE SIX OTHER
TABLES TO GET TO -

- AS FAST AS POSSIBLE, OF COURSE. WANT
SOMETHING ON TRAVELING SALESMAN?



why SAT?

- SAT 문제는 NP-Complete 문제의 본질이라고 할 수 있음
- 실질적인 논리 회로를 수학적으로 모델링해놓은 문제이기 때문!
- 때문에 대부분의 NP-Complete 문제는 SAT 문제로 환원할 수 있음 (사실상 모든이라고 봐도 되지만, 단언할 수 있는지는 모르겠음)
- knapsack -> sat? tsp -> sat? 환원 가능하지만, 난 못하니 생략 ㅎㅎㅎㅎ

Circuit SAT

- “실질적인 논리 회로를 수학적으로 모델링해놓은 문제이기 때문!”
- Circuit SAT은 이러한 본질을 가장 잘 표현해주는 문제
- “AND, OR 게이트로 이루어진 회로가 있을 때 회로의 결과를 참으로 만들어주는 입력이 있는가?”
- 예상했듯이, 이 문제도 3-SAT으로 환원 가능 (증명 x)

3-SAT을 풀기 위한 시도

- Brute Force로는 당연히 $O(2^m)$ 가량의 시간이 걸리나,
- $O(2^{0.386m})$ 의 알고리즘이 존재함. (현존하는 가장 빠른 알고리즘으로 알고 있음!)
- 이를 다항시간에 푼다면 NP-Complete를 P로 보낼 수 있지만, 안될거야 아마..
- 2-SAT은 이것보다는 단순한 형태지만 역시 다항시간 안에 풀릴 가능성은 희박하다..... 일 줄 알았지만..

2-SAT

- $(a \vee b) \wedge (c \vee d) \wedge (c \vee e) \wedge (a \vee e) = \text{TRUE}$
- 어떻게 풀까?
- 결론부터 말하자면 선형 시간에 풀린다.
- $O(\text{논리식 변수} + \text{논리식 개수})$

2-SAT

- 문제가 처음 제안되었을 때는 $O(n^4)$ 알고리즘이 발표되었으며 이후 $O(n)$ 으로 최적화되었다.
- 다만 구현 등 여러 면에서 상당히 복잡한 알고리즘이었다
- 하지만 1979년 Robert Tarjan이 힘써고 강한 그래프를 사용해서 선형 시간으로 한방에 날려버렸다. 찬양
- 2-SAT은 그래프로 변형된다고? 식의 성질을 살펴보자

Solving 2-SAT

- 논리식을 명제로 바꿔서 명제들의 집합으로 관리하는 게 핵심 아이디어이다.
- $a \vee b$ 가 있다고 치자, 이걸 a 가 참이면 무조건 참이고, a 가 거짓이면 b 가 참이어야 참이다.
- $a \rightarrow b$ 가 있다고 치자. a 가 거짓이면 가정이 거짓이므로 무조건 참이고, a 가 참이면 b 가 참이어야 참이다.
- ???
- $a \vee b = (\sim a \rightarrow b)$!!!!

Solving 2-SAT 2

- 이젠 증명 진짜 안하니까 궁금하면 Robert Tarjan님한테 물어봐라.
- 정점은 각 변수와 변수의 역 두개를 만들어준다
- $a \vee b \rightarrow (\sim a \rightarrow b) \ \&\& \ (\sim b \rightarrow a)$ 두개의 에지를 만든다
- 두 정점을 고려했을때 a에서 b로 갈수 있고 b에서 a로 갈수 있으면 둘은 같은 논리식이 적용되는 공동운명체다
- 그리고 a와 $\sim a$ 가 공동운명체면 그 SAT은 답이 없다

Solving 2-SAT 3

- 저대로만 해서 Floyd-Warshall 잘 쓰면 그걸로도 사실 잘 풀린다 $O(n^3)$ BFS $O(n^2)$ 는 생략
- 하지만 내가 아까 말한 공동운명체를 찾는 방법은 SCC를 사용하면 된다.
- 자세한 설명은 생략한다. (노란책 or 하얀책 참고)
- 이는 $O(V+E)$ 시간에 해결가능하다.
- 연습 문제 : <https://www.acmicpc.net/problem/2519>

Additional Note

- (Weighted) Maximum Independent Set 문제 역시 유명한 NP-Complete 문제
- 하지만 여러 신기한 그래프에서 이 문제를 다항 시간에 풀 수 있다. (난이도순)
 - 1) 그래프가 완전 그래프라면? $O(n)$
 - 2) 그래프가 순열 그래프라면? $O(n \lg n)$
 - 3) 그래프가 트리라면? $O(n)$ (KOI 예선기출, BOJ 2533)

more

- 4) 그래프가 이분 그래프라면? $O(n^{2.5})$
- 5) 그래프가 선분 그래프라면? $O(n \lg n)$
- 6) 그래프가 사다리꼴 그래프라면? $O(n \lg n)$ (Balkan 2011 기출, oj.uz balkan11_trapezoid)
- 7) 그래프가 이분인듯이분아닌이분같은 $\square \sqcup \circ \sqcup$ (IOI 2014 - Friend)

참고 문제

- 2SAT (<https://www.acmicpc.net/problem/2519>)
 - 풀이 (<http://amugelab.tistory.com/entry/%EB%A7%89%EB%8C%80%EA%B8%B0-KOI-2012>)
- 트리에서의 독립집합 (<https://www.acmicpc.net/problem/2533>)
- 선분 그래프에서의 독립집합 (http://koistudy.net/?mid=prob_page&NO=192)
- 사다리꼴 그래프에서의 독립집합 (http://oj.uz/problems/view/balkan11_trapezoid)
- 어떤 이상한 그래프에서의 독립집합 (http://oj.uz/problems/view/IOI14_friend)