CS 341 – Algorithms

Lecture 20 – Hard Partitioning Problems

30 July 2021

Today's Plan

1. 3-Dimensional Matching
2. Subset-Sum
3. Concluding Remarks

skip 3-coloring in L19

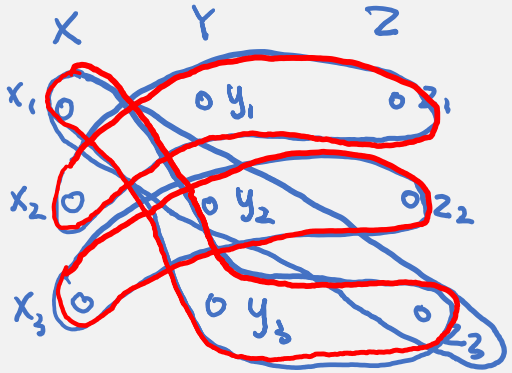
3-coloring won't be asked in HW and exam

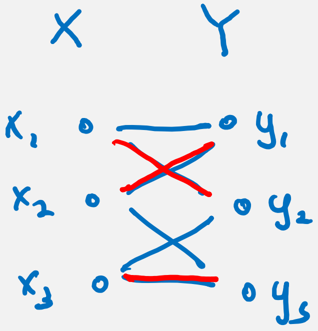
3-Dimensional Matching

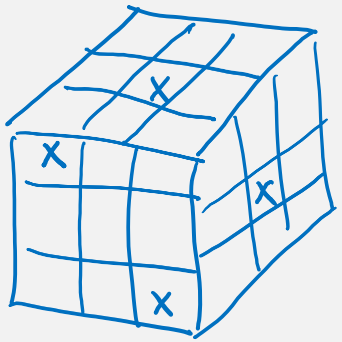
3-Dimensional Matching (3DM)

Input: Disjoint sets X, Y, Z, each of size n, a set $T \subseteq X \times Y \times Z$ of triples.

Output: Does there exist a subset of n disjoint triples in T? $\Leftrightarrow$ Cover every element exactly once.







$T = (x_1, y_2, z_3), (x_2, y_1, z_1)$
$(x_3, y_2, z_2), (x_1, y_3, z_3)$

A subset of n disjoint triples is called a perfect 3D-matching.

3DM is NP-complete

Theorem. 3DM is NP-complete.

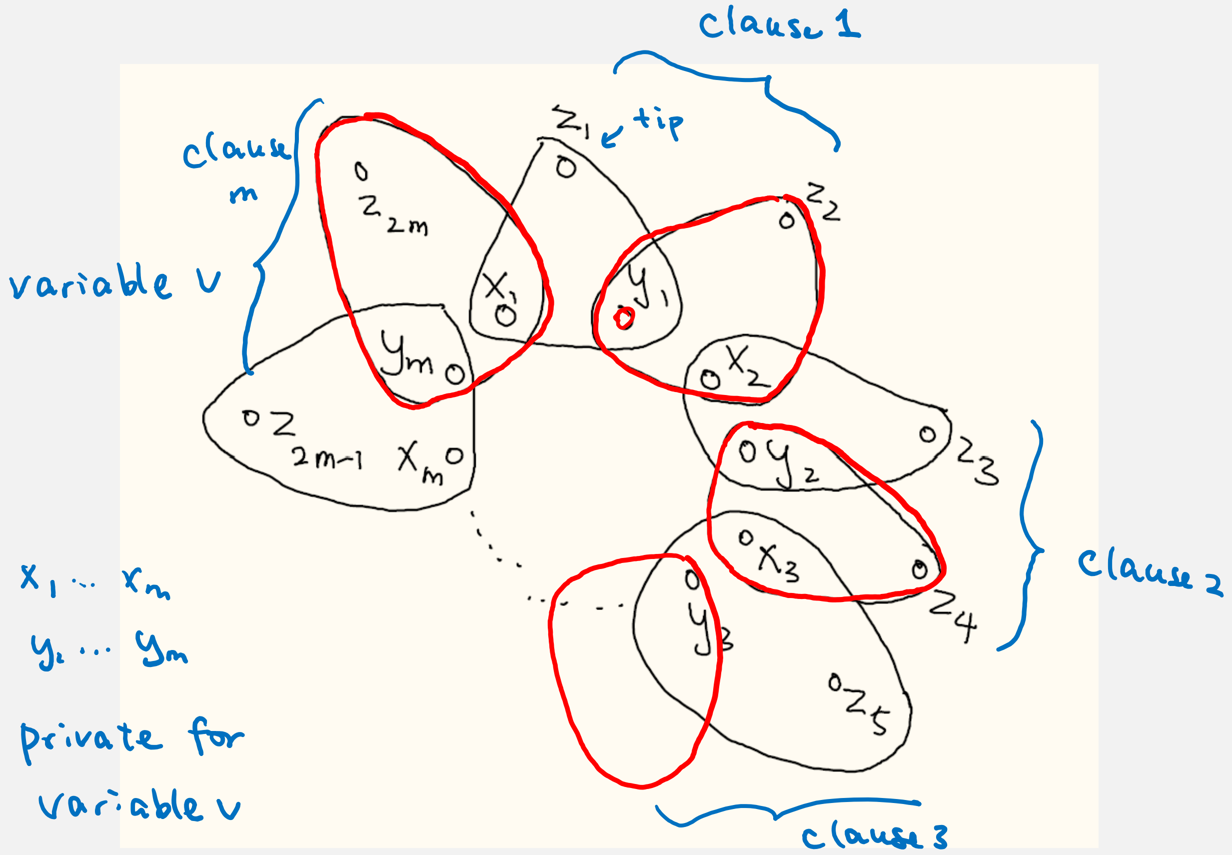
Proof. It is easy to check that 3DM is in NP.

To prove that it is NP-complete, we will prove that 3SAT $\leq_p$ 3DM.

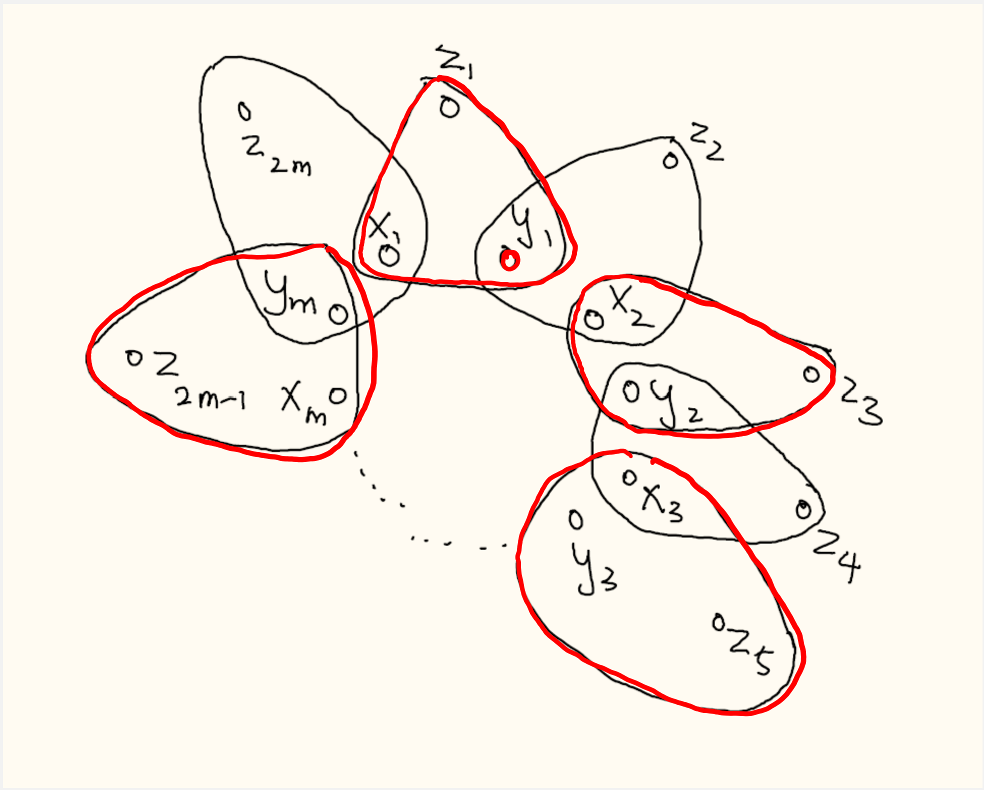
Given a 3SAT instance with n variables $x_1, x_2, \dots, x_n$ and m clauses $C_1, C_2, \dots, C_m$, we would like to construct a 3DM instance so that the formula is satisfiable iff there is a perfect 3D-matching.

Variable Gadget

We create some variable gadgets to capture the binary decision of a Boolean variable.



v = true ⇔ odd tips free

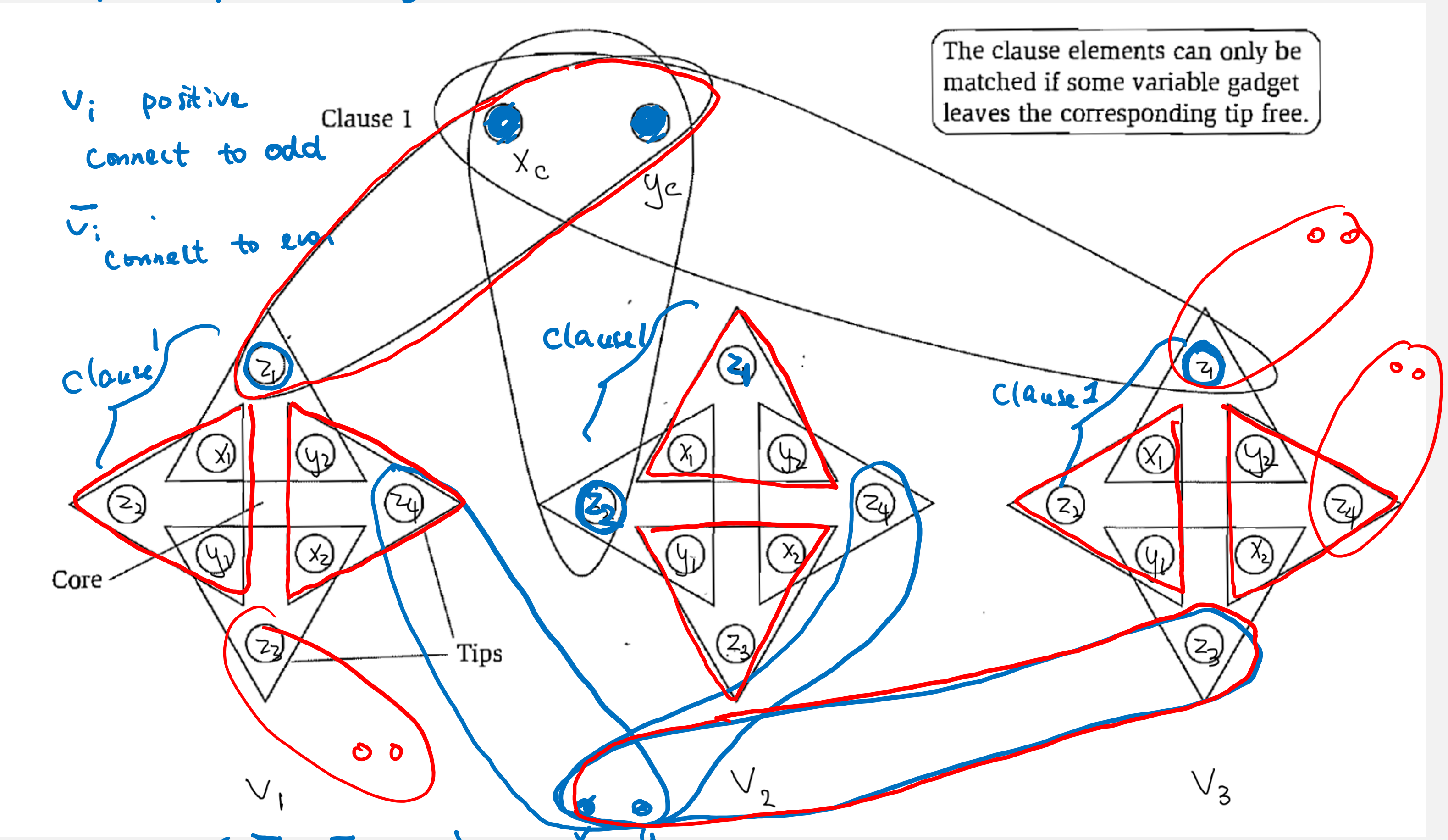


v = false ⇔ even tips open

Clause Structure

Now we add some clause structure to the 3DM instance so that only satisfying assignments "survive".

$C_1 = (v_1 \vee \bar{v}_2 \vee v_3)$



$C_2 = (\bar{v}_1 \vee \bar{v}_2 \vee v_3)$

$v_1 = T$

$v_3 = T$

Reduction

We create n variable gadgets, each with $2m$ “tips”.

Each clause has 3 different triples. Note that different clauses use different “tips” of the variable gadgets.

total tips = $2mn$

tips covered by clauses = m → $2mn - m = (2n-1)m$ tips left to be covered

add $(2n-1)m$ “dummy” clauses. add triples to each tip

Proof

Claim. The formula is satisfiable if and only if there is a perfect 3D-matching.

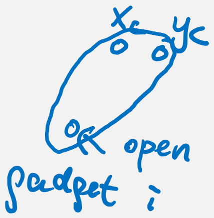
proof $\Rightarrow$ satisfying assignment

$x_i = T \Rightarrow$ gadget i leave odd tips open/free

$x_i = F \Rightarrow$ leave even tips free

for each clause, satisfied by some literal $C = (x_1 \cup x_2 \cup x_3)$

$\Rightarrow$ Corresponding tip is free in that gadget T



for the remaining tips, use dummy to form a perfect 3DM.

□

Proof

Claim. The formula is satisfiable if and only if there is a perfect 3D-matching.

proof $\Leftarrow$ perfect 3DM

in each gadget. either all even tips open or all odd tips open

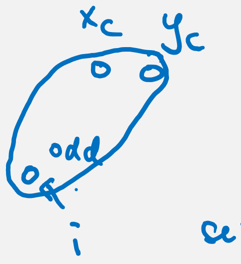
all odd tips open inside gadget i $\rightarrow$ $x_i = T$

even $\rightarrow$ $x_i = F$

perfect 3DM $\rightarrow$ clause elements be covered

$\rightarrow$ asg satisfy the clause

based on our construction



$C = (v_i \cup v_j \cup v_k)$

set $v_i = T$

$\rightarrow$ every clause is satisfied. $\square$

Today's Plan

1. 3-Dimensional Matching

2. Subset-Sum

3. Concluding Remarks

Subset-Sum

Input: n positive integers $a_1, a_2, \dots, a_n$ and an integer K.

Output: Does there exist a subset $S \subseteq [n]$ with $\sum_{i \in S} a_i = K$?

Theorem. Subset-Sum is NP-complete.

Proof. It is easy to check that Subset-Sum is in NP.

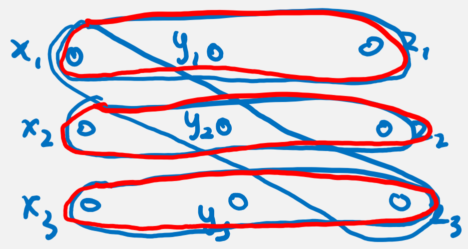
To prove that it is NP-complete, we will prove that 3DM $\leq_p$ Subset-Sum.

Given a 3DM instance, we would like to construct a Subset-Sum instance so that there is a perfect 3D-matching if and only if there is a subset of certain sum K (value to be determined later).

(It requires some new idea to solve a problem about numbers.)

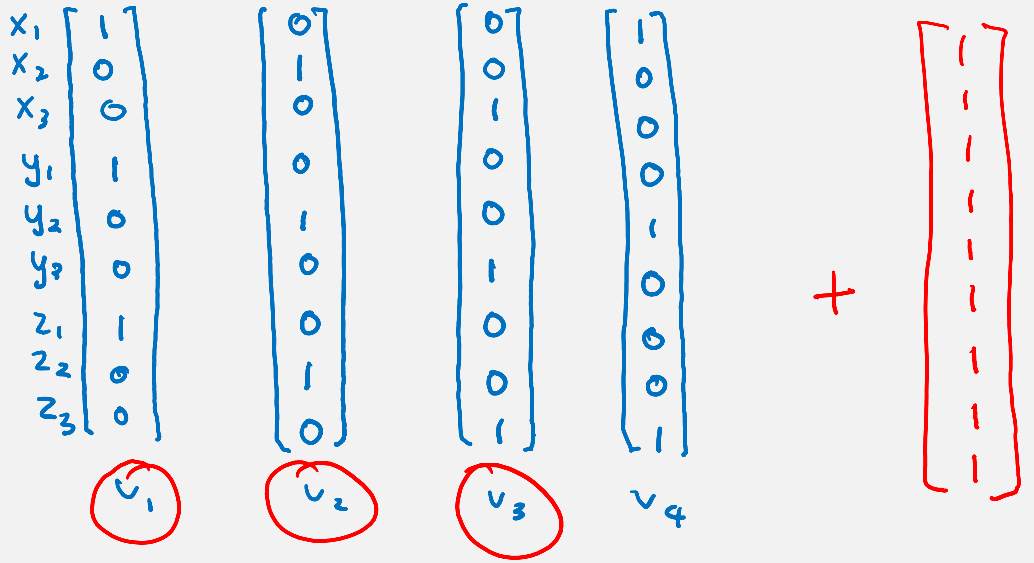
Vector Representation of 3DM

To see the connection between 3DM and Subset-Sum, it is easier to use a different way to see 3DM.



$t_1 = (x_1, y_1, z_1)$ $t_2 = (x_2, y_2, z_2)$

$t_3 = (x_3, y_3, z_3)$ $t_4 = (x_1, y_2, z_3)$



Claim: $\exists$ a perfect 3DM

$\Leftrightarrow$ $\exists$ a subset of vectors sums to $\vec{1}$

vector subset-sum

Input: a set of n 0-1 vectors

Output: does there exist a subset of vectors that sum to $\vec{1}$?

Reduction Idea

A very natural idea is to interpret the 0-1 vector as the binary representation of a number.

$x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3$: $\begin{bmatrix}1\\0\\0\\0\\0\\1\\0\\0\\1\end{bmatrix}$ $\rightsquigarrow$ 1 0 0 1 0 0 0 0 1

$z_3\ z_2\ z_1\ y_3\ y_2\ y_1\ x_3\ x_2\ x_1$

vector $\rightsquigarrow$ number

$(x_i, y_j, z_k) \rightsquigarrow 2^i + 2^{n+j} + 2^{2n+k}$

<u>Claim</u> If $\exists$ a perfect 3DM, then there is a subset with sum $\sum_{l=1}^{3n} 2^l$.

$\nLeftarrow$ vector addition $\neq$ number addition (carrying)

$$\begin{array}{r} 01 \\ 01 \\ +\ 01 \\ \hline 11 \end{array}$$

Actual Reduction

There is a simple trick to get around this "carrying" problem, so that the above plan would work.

Imagine 9 triples, mapping each vector to a decimal number

$(z_3, z_2, z_1, y_3, y_2, y_1, x_3, x_2, x_1)$ — record # of times element got covered.

e.g.

m triples, map each 0-1 vector to a base (m+1)-number

$$(x_i, y_j, z_k) \rightsquigarrow (m+1)^i + (m+1)^{n+j} + (m+1)^{2n+k}$$

this is the reduction

$$K = \sum_{l=1}^{3n} (m+1)^l$$

Proof

$b = (m+1)$

Claim. There is a perfect 3D-matching if and only if there is a subset with sum $K = \sum_{i=1}^{3n} b^i$.

proof $\Rightarrow$ done.

$\Leftarrow$ because $b = m+1$ large enough

each "coordinate" records how many times an element is covered.

to achieve sum K, each element must be covered exactly one $\leadsto$ perfect 3DM.

□

Corollary. Knapsack is NP-complete.

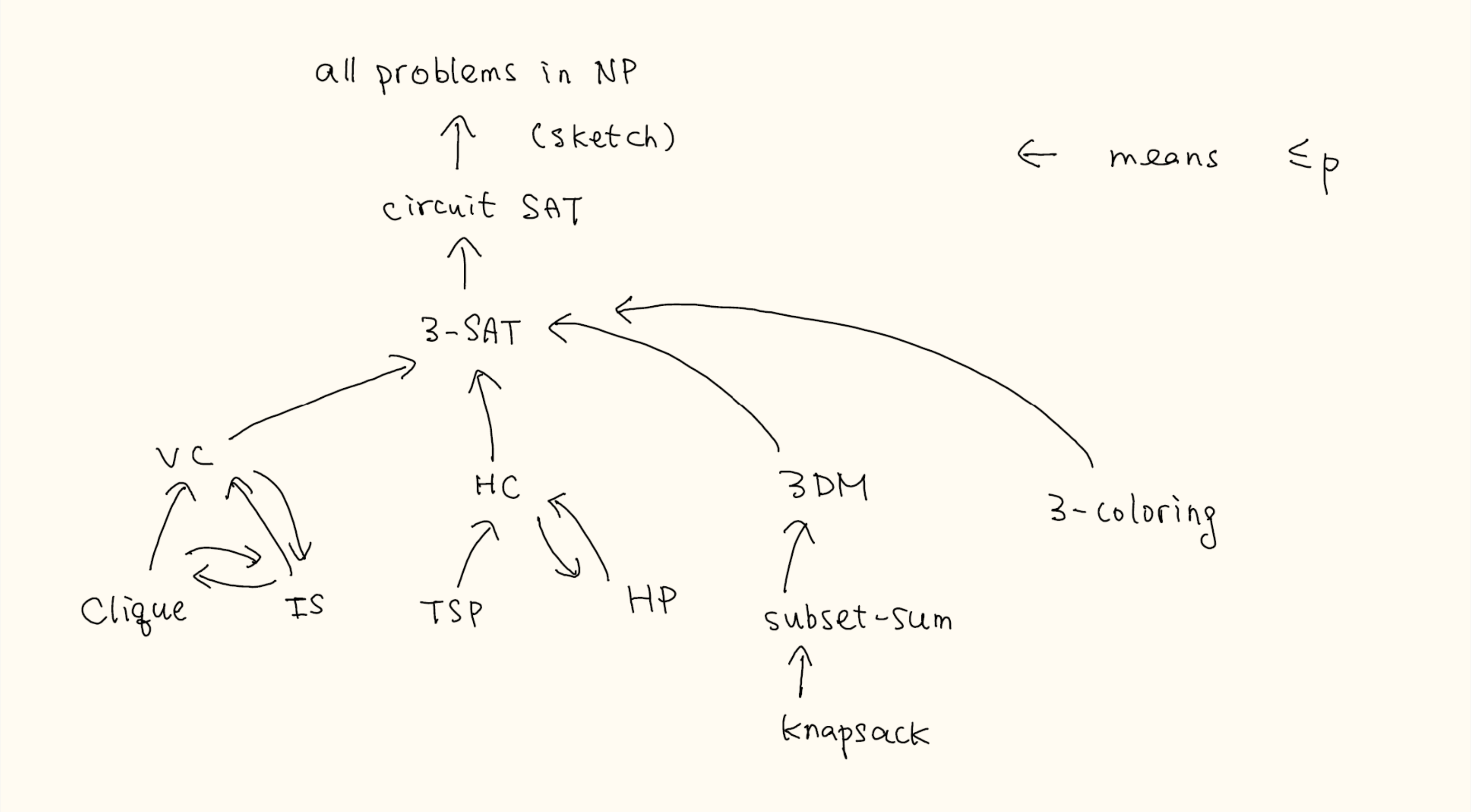
Today's Plan

1. 3-Dimensional Matching

2. Subset-Sum

3. Concluding Remarks

Map



In NP-completeness, we showed you the difficult reductions and ask you to do the easier ones.

You can only use problems from this map to do homework and exam.

Techniques for Doing Reductions

It requires practices to search for the right problem Y to prove Y $\leq_p$X for our problem X.

There are three common techniques in proving NP-completeness.

Specialization: subset-sum $\leq_p$ knapsack

✓ HC $\leq_p$ TSP

HP $\leq_p$ bounded degree spanning tree

Local Replacement: circuit-SAT $\leq_p$ 3-SAT

✓ 3DM $\leq_p$ vector subset-sum $\leq_p$ subset-sum

DHC $\leq_p$ HC

Gadget Design: 3SAT $\leq_p$ DHC 3SAT $\leq_p$ 3-coloring

X 3SAT $\leq_p$ 3DM

2 vs 3

$x_i \vee x_j$

$\neg x_i \to x_j$

2 SAT
↑
polytime

3 SAT

NPC

2 DM
↑
polytime

3DM

NPC

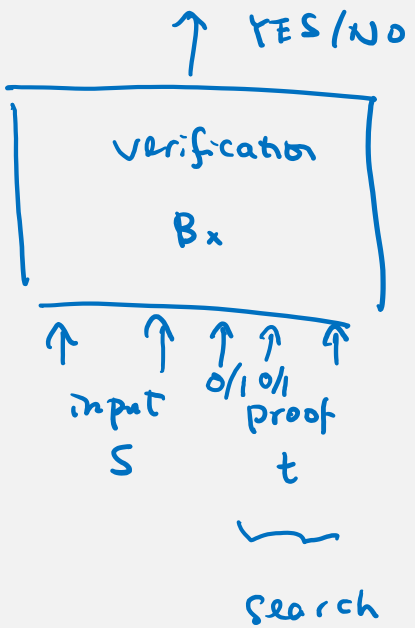
2 coloring

P

3 coloring

NPc

Decision Problems vs Search Problems



reconstruct the solution bit by bit by using the decision algorithm.

Final Exam

Good luck in your final exam.

1. focus on second half (greedy . DP. bipartite matching . NPc)
2. harder than midterm (topics . difficulty)
3. answer shorter !
4. only ask clarification questions
5. piazza post early next week

Learning Outcome (from L01)

- Know basic techniques and well-known algorithms well.
- Have the skills to design new algorithms for simple problems.
- Have the skills to prove correctness and analyze time complexity of an algorithm.
- Use reductions to solve problems and to prove hardness.

recursions

What is Ahead

- Probability: Randomized Algorithms, Probabilistic Methods, Random Sampling
- Linear Algebra: Linear Equations, Matrix Algorithms, Spectral Methods
- Optimization (Calculus): Linear Programming, Approximation Algorithms, Convex Optimization
- Complexity: Hardness of Approximation, Fine-Grained Hardness