# CS 341: Algorithms
# Lec 18: NP-completeness part 3

Armin Jamshidpey     Collin Roberts

Based on lecture notes by Éric Schost, and many other CS 341 instructors

David R. Cheriton School of Computer Science, University of Waterloo

Winter 2025

# Perfect 3D matchings

## 3D matching

- **input:** 3 sets $X, Y, Z$ of size $n$ and a family of **hyperedges** $E \subset X \times Y \times Z$
- **output:** is there a **perfect matching** ($n$ hyperedges that cover $X$, $Y$ and $Z$)?

  (each $x_i$ (and each $y_j$, and each $z_k$) is in a unique hyperedge)
- **NP**

## Remark: 2D version

- **input:** 2 sets $X, Y$ of size $n$ and a family of **edges** $E \subset X \times Y$
- **output:** is there a **perfect matching** ($n$ edges that cover $X$, $Y$)?
- this is testing if a bipartite graph has a perfect matching

# 3DMatchings is NP-complete

**Claim**

3SAT $\leq_P$ 3DMatching

- **given:** a formula $F$ in 3CNF, with $s$ clauses $C_1, \ldots, C_s$
- **want:** build an instance $H$ of 3DMatching such that $F$ satisfiable iff $H$ admits a perfect 3D matching
- reduction must be polynomial time

# The variable gadget

build **one fidget-spinner-thing per variable** $x_i$, $i = 1, \ldots, n$.

**Vertices**

- $2s$ **core vertices** $v_{i,1}, \ldots, v_{i,2s}$ only used in the gadget
- $2s$ **tip vertices** $z^T_{i,1}, z^F_{i,1}, \ldots, z^T_{i,s}, z^F_{i,s}$ will connect to clause vertices

**Hyperedges** for $j = 1, \ldots, s$

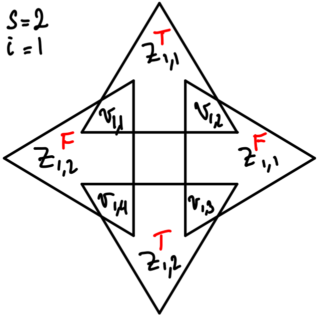
- $z^T_{\boldsymbol{i,j}}, v_{\boldsymbol{i,2j-1}}, v_{\boldsymbol{i,2j}}$
- $z^F_{\boldsymbol{i,j}}, v_{\boldsymbol{i,2j}}, v_{\boldsymbol{i,2j+1}}$

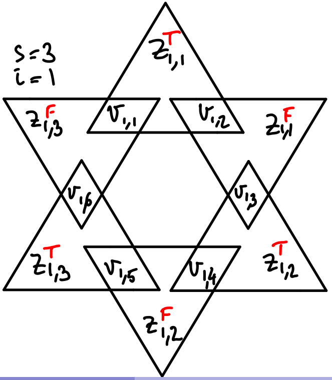
# The variable gadget

build **one fidget-spinner-thing per variable** $x_i$, $i = 1, \ldots, n$.

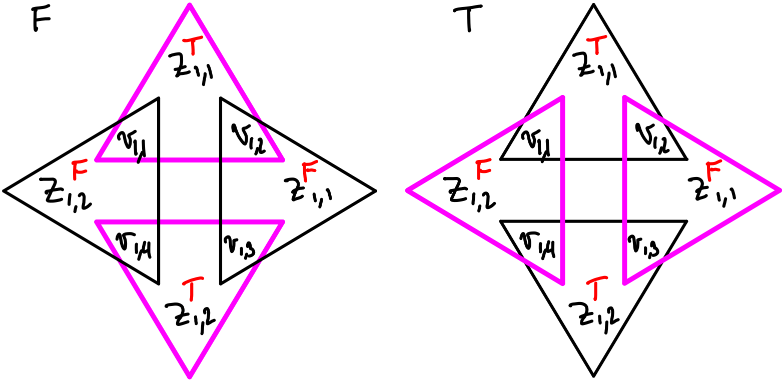
### Vertices

- $2s$ **core vertices** $v_{i,1}, \ldots, v_{i,2s}$ only used in the gadget
- $2s$ **tip vertices** $z^T_{i,1}, z^F_{i,1}, \ldots, z^T_{i,s}, z^F_{i,s}$ will connect to clause vertices



# Covering the core vertices

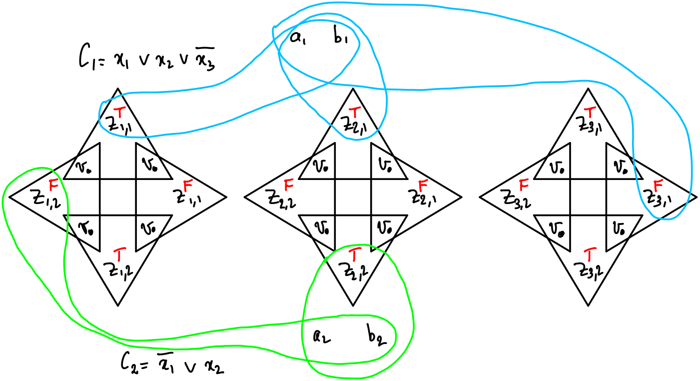



$2^n$ coverings (2 possibilities per variables)

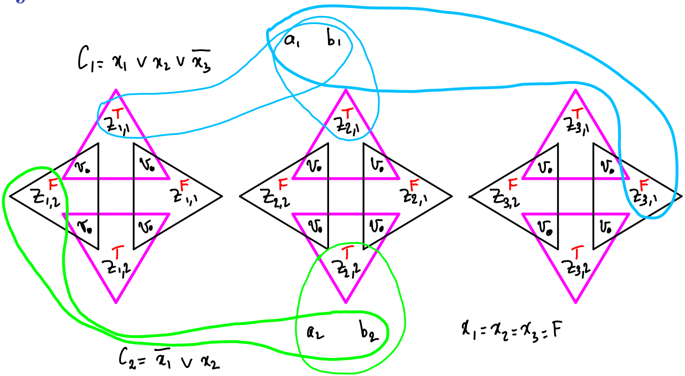
# Using the clauses to (almost) finish the graph

**For any clause $C_j$**

- add two new vertices $a_j$ and $b_j$
- for any literal $\boldsymbol{x_i}$ in $C_j$, add hyperedge $\boldsymbol{\{a_j, b_j, z^T_{i,j}\}}$
- for any literal $\overline{\boldsymbol{x_i}}$ in $C_j$, add hyperedge $\boldsymbol{\{a_j, b_j, z^F_{i,j}\}}$



# Final adjustments




- we have $2ns$ tips (in the example it is 12)
- in a perfect matching, each clause covers a tip, so $2ns - s$ tips left (in the example it is 10)
- in a perfect matching, $ns$ tips will be covered by hyperedges which cover core vertices (triangle hyperedges).
- we add $ns - s$ dummy pairs $d_k, e_k$ and **all** hyperedges $\{z_{i,j}^T, d_k, e_k\}$ and $\{z_{i,j}^F, d_k, e_k\}$ (that's $(ns - s)(2ns)$ )

# $F$ satisfiable iff perfect 3D matching

## If $F$ is satisfiable

- cover gadgets for $x_1, \ldots, x_n$ according to their truth value
- pick **exactly** one true literal per clause $C_j$
  - if $x_i$, take hyperedge $\{a_j, b_j, z_{i,j}^T\}$
  - if $\overline{x_i}$, take hyperedge $\{a_j, b_j, z_{i,j}^F\}$
- match all remaining tips with pairs of dummy vertices

## If perfect 3D matching

- matching gives truth values
- for each clause $C_j$, we picked a hyperedge $\{a_j, b_j, z_{i,j}^T\}$, resp. $\{a_j, b_j, z_{i,j}^F\}$
- the corresponding $x_i$ is $T$, resp. $F$
- this makes $C_j$ satisfied either way

# Subset sum is NP-complete

**Subset sum**

- **given:** positive integers $a_1, \ldots, a_n$ and $K$
- **want:** is there a subset $S$ of $\{1, \ldots, n\}$ with $\sum_{i \in S} a_i = K$
- **NP**

**Claim**

$\textsc{3DMatching} \leq_P \textsc{SubsetSum}$

- **given:** sets $X, Y, Z$ of size $n$, $m$ hyperedges $E \subset X \times Y \times Z$
- **want:** integers $a_1, \ldots, a_s, K$ s.t. perfect 3D matching iff $\sum_{i \in S} a_i = K$ for some $S \in \{1, \ldots, n\}$
- reduction must be polynomial time

# From 3D matchings to vectors

we define $\boldsymbol{m}$ 0/1 vectors (one per hyperedge) of size $3n$.

- $j$th hyperedge $= \{x_{\boldsymbol{u}}, y_{\boldsymbol{v}}, z_{\boldsymbol{w}}\}$, $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$ in $\{1, \ldots, n\}$
- $j$th vector given by

$$v_j = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$\uparrow \; u \qquad \uparrow \; v+n \qquad \uparrow \; w+2n$$

**Observation**

there is a perfect 3D matching iff there is a subset of $\{v_1, \ldots, v_m\}$ that adds up to

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdots & \cdots & \cdots & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

# From vectors to numbers

we define $\boldsymbol{m}$ integers (one per hyperedge) of bit-size polynomial in $n, m$

- set $\boldsymbol{b = m + 1}$
- given

$$v_j = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$\uparrow \qquad\qquad\qquad \uparrow \qquad\qquad\qquad \uparrow$$

$$u \qquad\qquad\qquad v+n \qquad\qquad\qquad w+2n$$

define

$$\boldsymbol{a_j = b^u + b^{v+n} + b^{w+2n}}$$

(vector $v_j \leftrightarrow$ digits of $a_j$ in base $b$)

- $a_j \le (m+1)^{3n+1}$ so $\log_2(a_j) \in (mn)^{O(1)}$

# End of the proof

set $K = b + b^2 + \cdots + b^{3n}$

**Claim**

for $S$ subset of $\{1, \ldots, m\}$, $\sum_{i \in S} v_i = [1 \ \cdots \ 1] \iff \sum_{i \in S} a_i = K$

1. we always have

$$\sum_{i \in S} a_i = \sum_{j=1}^{3n} c_j b^j$$

with $c_j$ = number of $v_i$'s in $S$ with $v_{i,j} = 1$

2. if $\sum_{i \in S} v_i = [1 \ \cdots \ 1]$, $c_j = 1$ for all $j$, so $\sum_{i \in S} a_i = \sum_{j=1}^{3n} b^j = K$
3. if $\sum_{i \in S} a_i = K$,

$$\sum_{j=1}^{3n} b^j = \sum_{j=1}^{3n} c_j b^j$$