Representative Families and Algorithms

Lecture 3

Fedor V. Fomin

School on Algorithms,
Combinatorics, Complexity 2021

UNIVERSITY OF BERGEN

Week plan

- Bollobas Lemma
- Some combinatorial applications (critical graphs and minimal separators)
- Representative Sets
- Few attempts to compute Reps
- Longest Path application
- Matroids
- One more attempt to compute Reps
- Kernelization application

Recap from the yesterday's lecture

Let $\mathcal{F}$ be a family of a-sets, $\mathcal{F}' \subseteq \mathcal{F}$ b-represents $\mathcal{F}$ if for every B of size b such that there exists an $A \in \mathcal{F}$ with $A \cap B = \emptyset$ there exists an $A' \in \mathcal{F}'$ with $A' \cap B = \emptyset$.

Corollary of Bollobás: For every $\mathcal{F}$ there is an $\mathcal{F}' \subseteq \mathcal{F}$ of size at most $\binom{a+b}{b}$ that b-represents $\mathcal{F}$.

Recap from the yesterday's lecture

Let F be a family of a-sets, $F' \subseteq F$ b-represents F if for every B of size b such that there exists an $A \in F$ with $A \cap B = \emptyset$ there exists an $A' \in F'$ with $A' \cap B = \emptyset$.

Corollary of Bollobás: For every F there is an $F' \subseteq F$ of size at most $\binom{a+b}{b}$ that b-represents F.

Algorithm computing b-representative set

Output size: $2^{a+b+o(a+b)} \log n$

Running time: $|F| 2^{a+b} \log n$

Longest Path: Given a (directed) graph G and integer k, decide whether G contains a path with at least k vertices?

Longest Path: Given a (directed) graph G and integer k, decide whether G contains a path with at least k vertices?

Output size: $2^{a+b+o(a+b)}\log n$

Running time: $|F|2^{a+b}\log n$

Longest Path: Given a (directed) graph G and integer k, decide whether G contains a path with at least k vertices?

Output size: $2^{a+b+o(a+b)} \log n$

Running time: $|F| 2^{a+b} \log n$

Algorithm solving Longest Path in time $4^{k+o(k)} \cdot poly(n)$

Longest Path story

- Monien [1982], $k^{k} \cdot n^{O(1)}$ representative sets

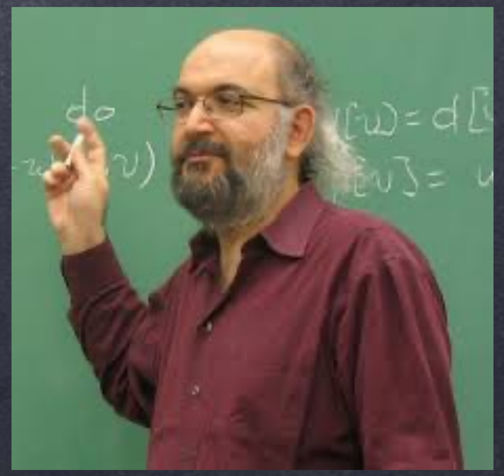
Burkhard Monien

- Bodlaender [1984]: $k^{k} \cdot n^{O(1)}$ treewidth

Hans Bodlaender

- Papadimitriou and Yannakakis [1996]: Is in P for $k = \log n$?

Christos Papadimitriou



Mihalis Yannakakis

Color Coding [1995] $O(2^{O(k)} \cdot n)$

Color-Coding

NOGA ALON

Institute for Advanced Study, Princeton, New Jersey and Tel-Aviv University, Tel-Aviv, Israel

RAPHAEL YUSTER AND URI ZWICK

Tel-Aviv University, Tel-Aviv, Israel

Abstract. We describe a novel randomized method, the method of *color-coding* for finding simple paths and cycles of a specified length k, and other small subgraphs, within a given graph $G = (V, E)$. The randomized algorithms obtained using this method can be derandomized using families of *perfect hash functions*. Using the color-coding method we obtain, in particular, the following new results:

—For every fixed k, if a graph $G = (V, E)$ contains a simple cycle of size *exactly* k, then such a cycle can be found in either $O(V^{\omega})$ expected time or $O(V^{\omega} \log V)$ worst-case time, where $\omega < 2.376$ is the exponent of matrix multiplication. (Here and in what follows we use V and E instead of $|V|$ and $|E|$ whenever no confusion may arise.)

—For every fixed k, if a *planar* graph $G = (V, E)$ contains a simple cycle of size *exactly* k, then such a cycle can be found in either $O(V)$ expected time or $O(V \log V)$ worst-case time. The same algorithm applies, in fact, not only to planar graphs, but to any *minor closed* family of graphs which is not the family of all graphs.

—If a graph $G = (V, E)$ contains a subgraph isomorphic to a *bounded tree-width* graph $H = (V_H, E_H)$ where $|V_H| = O(\log V)$, then such a copy of H can be found in *polynomial time*. This

Longest Path Story





Algebraic fingerprints

Determinant-sum





Treewidth algorithms





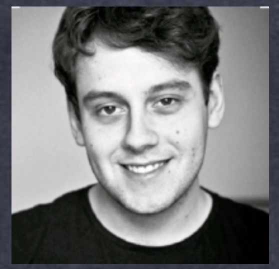
Divide-and-color

Representative sets



Cut & count



Narrow sieves

Polynomial differentiation

Dynamic Programming Held-Karp, Bellman [1962]: $O(2^n)$

	1	2	3	…	i	…	n
v_1							
v_2							
v_3							
…							
v_j							
…							
v_n							

Richard Karp

Richard Bellman

Dynamic Programming Held-Karp, Bellman [1962]: $O(2^n)$

	1	2	3	...	i	...	n
v_1							
v_2							
v_3							
...							
v_j							
...							
v_n							

$F[v_j, i]$ = [Vertex Sets of paths of length i ending in v_j]

Richard Karp

Richard Bellman

Dynamic Programming Held-Karp, Bellman [1962]: $O(2^n)$

	1	2	3	…	i	…	n
v_1							
v_2							
v_3							
…							
v_j							
…							
v_n							

$F[v_j, i]$ = [Vertex Sets of paths of length i ending in v_j]

Sets, not sequences!!!

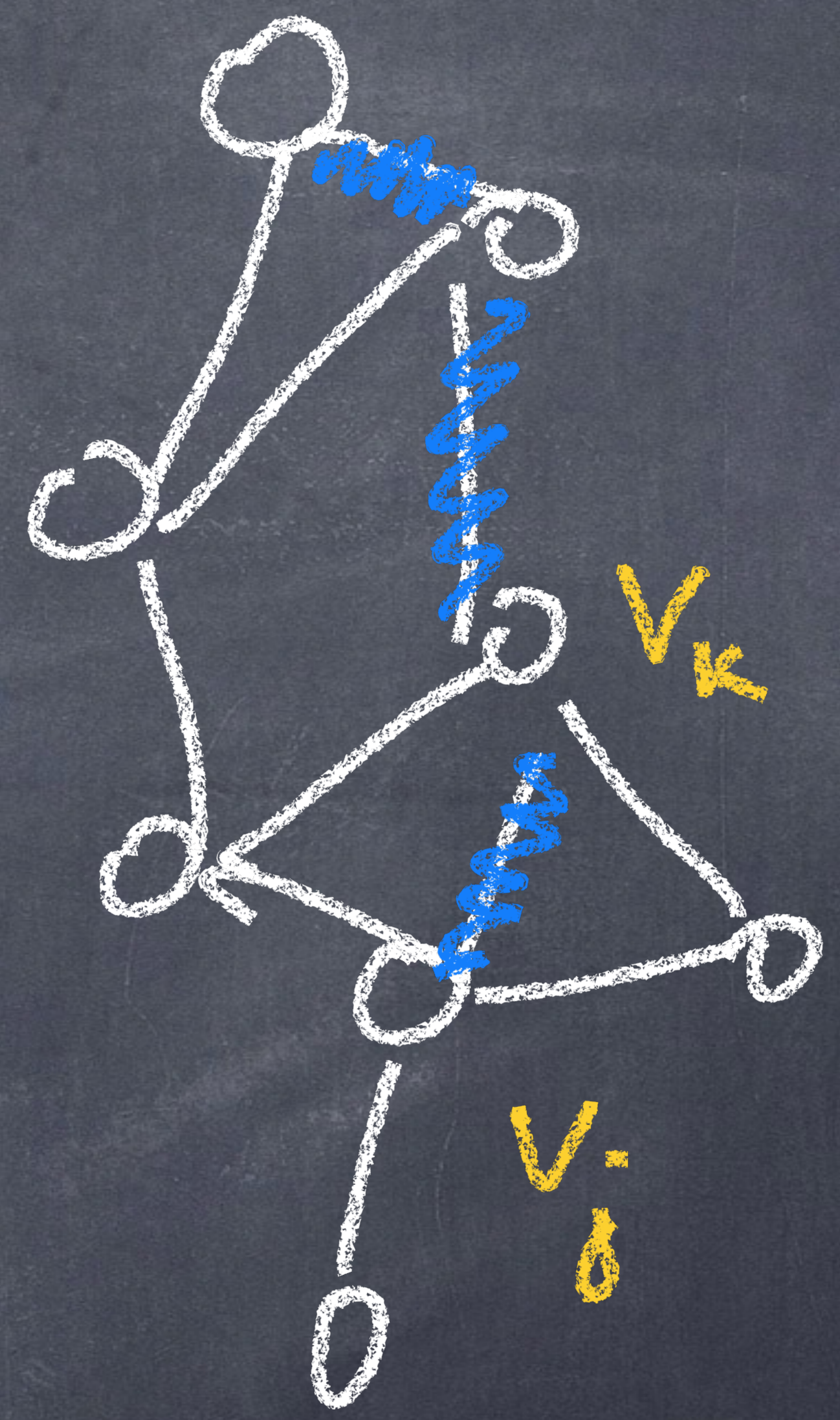
Richard Karp

Richard Bellman

Dynamic Programming Held-Karp, Bellman [1962]: $O(2^n)$

	1	2	...	i-1	i	...	n
v_1							
v_2							
v_3							
...							
v_j	$F[v_j, 1]=\{v_j\}$						
...							
v_n							

$F[v_j,i]$=[Vertex Sets of i-paths ending in v_j]



Dynamic Programming Held-Karp, Bellman [1962]: $O(2^n)$

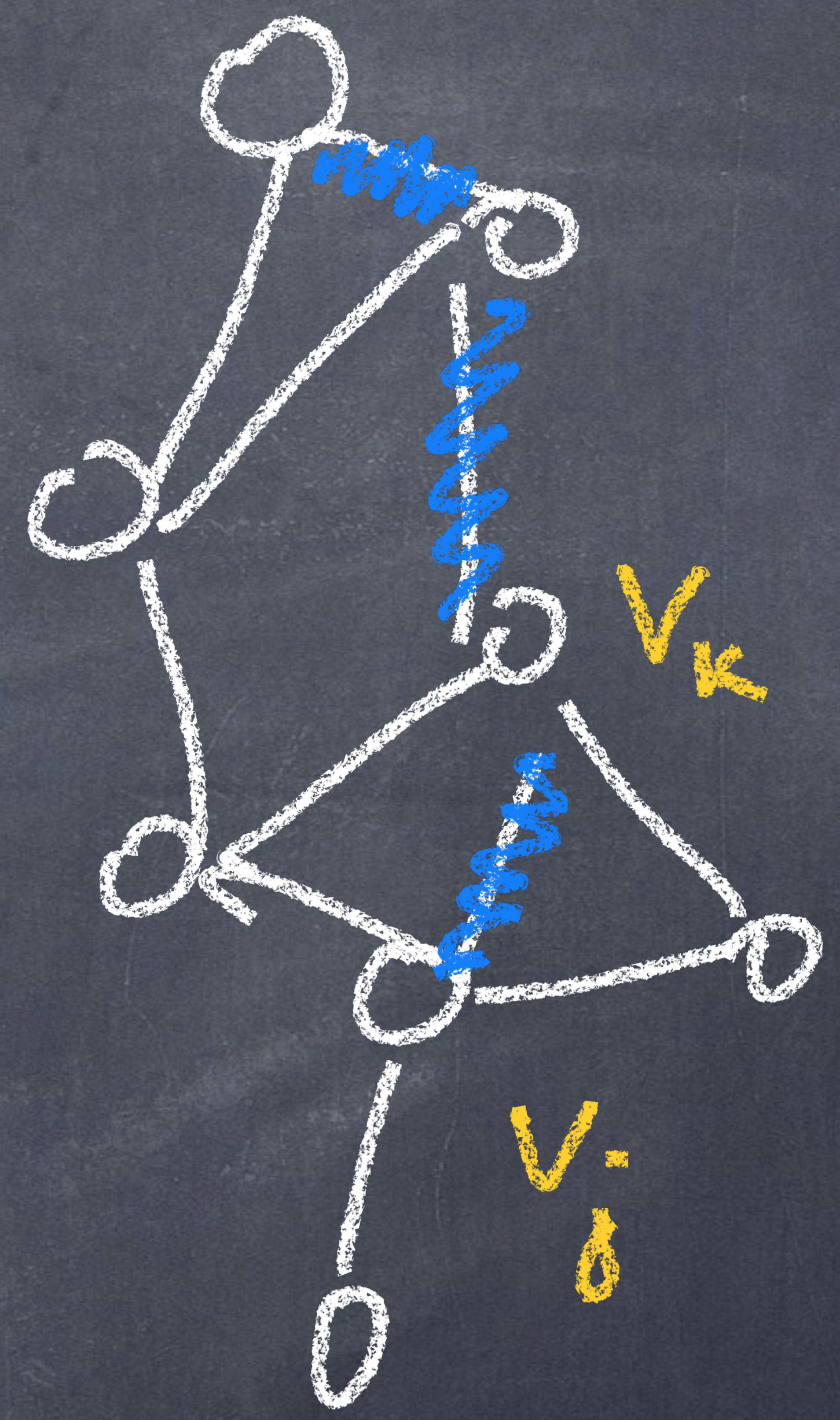
	1	2	…	i-1	i	…	n
v_1							
v_2							
v_3							
…							
v_j	$F[v_j, 1]=\{v_j\}$						
…							
v_n							

All $(i-1)$-paths that can be extended by v_j

$F[v_j,i]$=[Vertex Sets of i-paths ending in v_j]

=[Vertex Sets of $(i-1)$-paths avoiding v_j ending in

$v_k \in N(v_j) \cup \{v_j\}$]



Dynamic Programming for k-Path

	0	1	2	...	k	...	n
v_1							
v_2							
v_3							
...							
v_j							
...							
v_n							

k-Path, keep at most $\binom{n}{k}$ sets. Update time for each set polynomial

Dynamic Programming for k-Path

	0	1	2	…	k	…	n
v_1							
v_2							
v_3							
…							
v_j							
…							
v_n							

Time $n^{O(k)}$

k-Path, keep at most $\binom{n}{k}$ sets. Update time for each set polynomial

Dynamic Programming for k-Path, Reps enter the game

Time $n^{O(k)}$

Keep $\binom{n}{k}$ sets

Update time $\binom{n}{k} n$

Reps →

Time $4^{k+o(k)} \cdot n^{O(1)}$

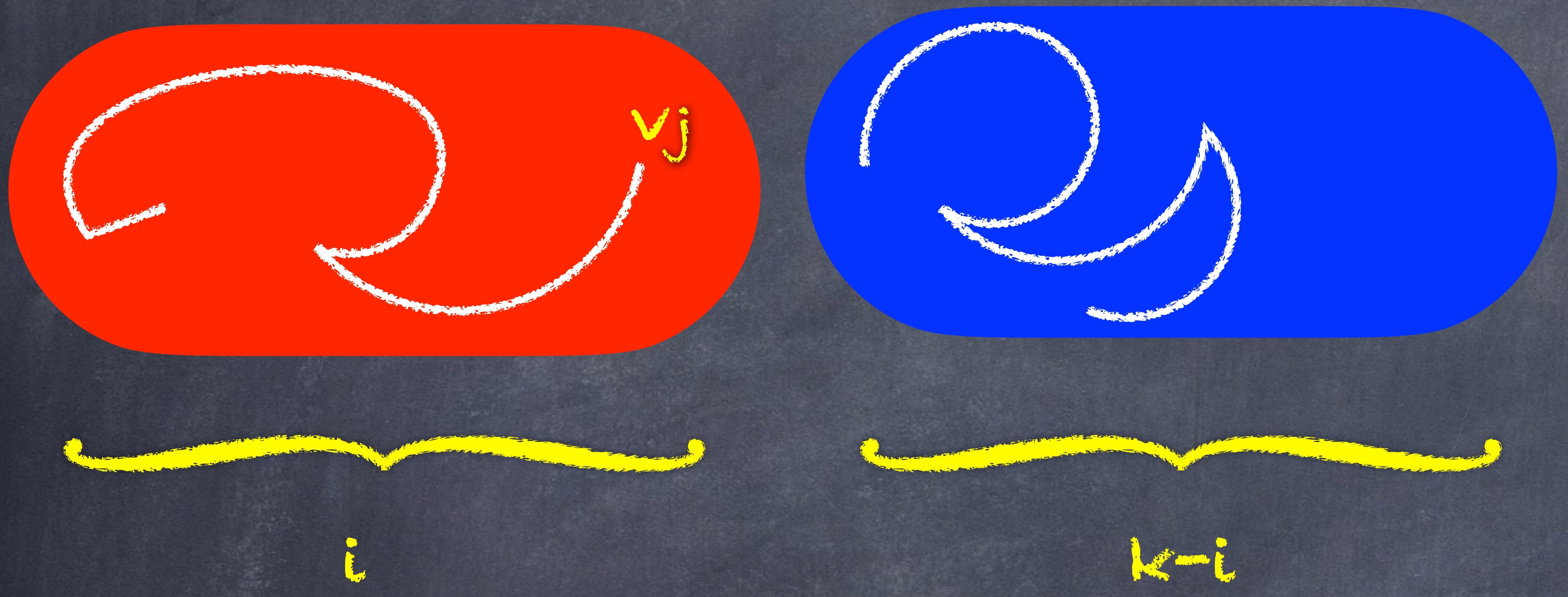
Keep $2^{k+o(k)} \log n$ sets

Update time

$2^k 2^{k+o(k)} n \log n = 4^{k+o(k)} \cdot n^{O(1)}$

Dynamic Programming for k-Path with Reps

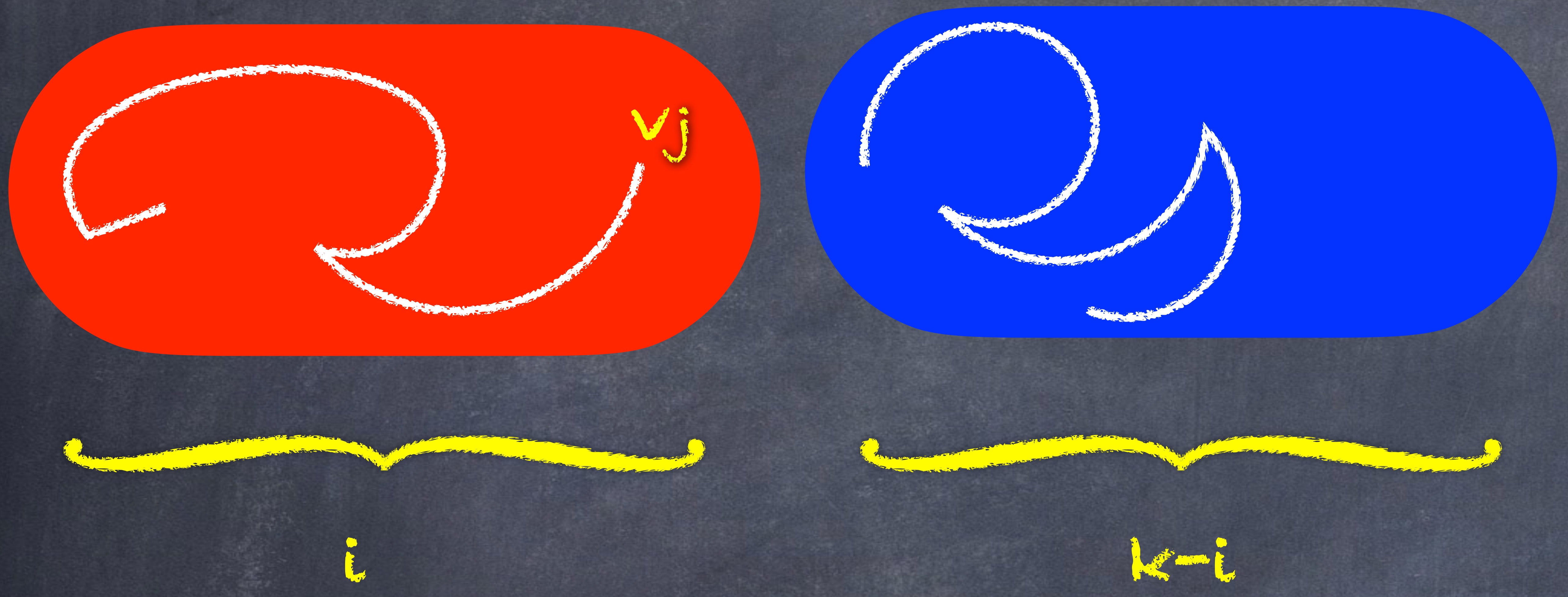
F[vj,i] All i-paths that end with vj



Dynamic Programming for k-Path with Reps

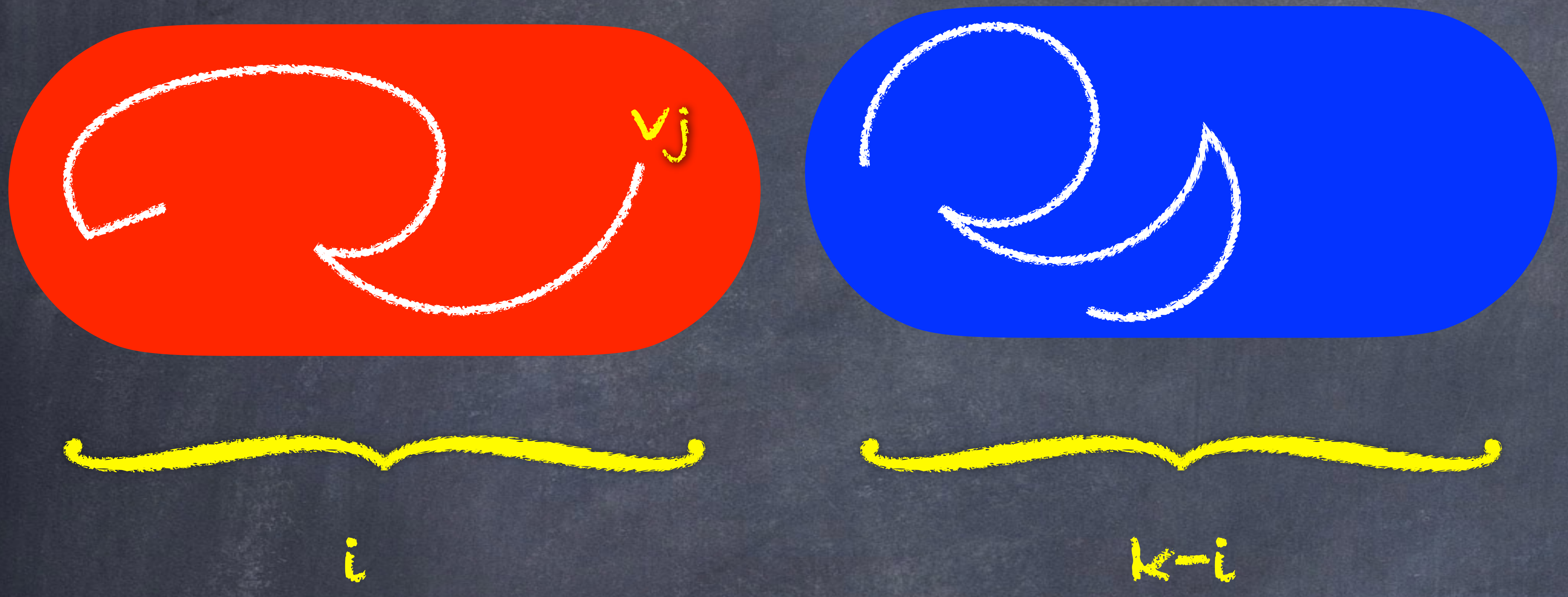
$F[v_j,i]$ All i-paths that end with v_j

$F'[v_j,i]$ family $(k-i)$-representing $F[v_j,i]$



Dynamic Programming for k-Path with Reps

F[vj,i] All i-paths that end with vj



F'[vj,i] family (k-i)-representing F[vj,i]

↓ Reps

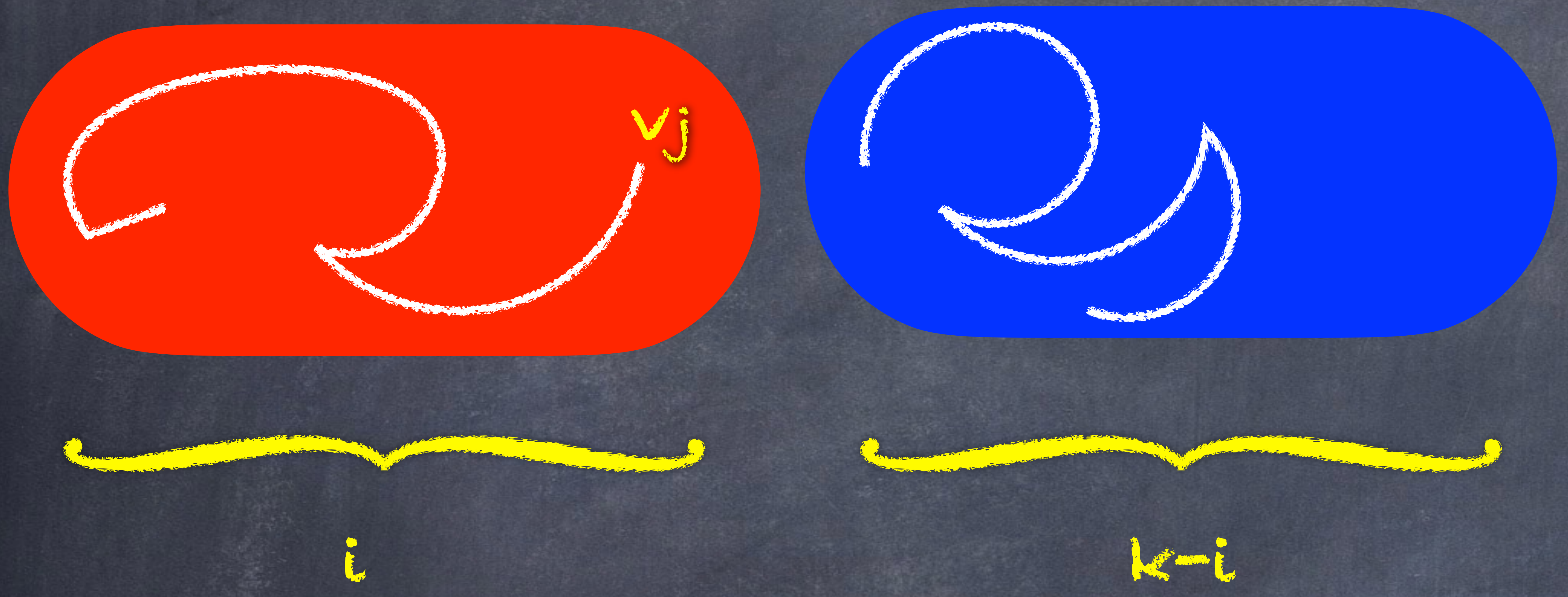
Output size: $2^{k+o(k)} \log n$

Running time:

$$|F[v_j, i]|\,2^k \log n = \binom{n}{k} 2^k \log n$$

Dynamic Programming for k-Path with Reps

F[vj,i] All i-paths that end with vj



F'[vj,i] family (k-i)-representing F[vj,i]

↓ Reps

Output size: $2^{k+o(k)} \log n$

Running time:

$$|F[v_j, i]|\, 2^k \log n = \binom{n}{k} 2^k \log n$$

Time $n^{O(k)}$

not what we shoot for!

Dynamic Programming for k-Path with Reps

$F'[v_j,1]=\{v_j\}$

Dynamic Programming for k-Path with Reps

$F'[v_j,1]=\{v_j\}$

$$X[v_j, i] = \bigcup_{v_k v_j \in E(G)} F'[v_k, i-1] \cup \{v_j\}$$

Sets not containing v_j

Dynamic Programming for k-Path with Reps

F'[vj,1]={vj}

$$X[v_j, i] = \bigcup_{v_k v_j \in E(G)} F'[v_k, i-1] \cup \{v_j\}$$

Sets not containing vj

F'[vj,i]=REDUCE(X[vj,i])

Output size: $2^{k+o(k)} \log n$

Running time: $|X[v_j, i]|\, 2^k \log n = 2^{k+o(k)} n 2^k \log n$

Dynamic Programming for k-Path with Reps

F'[vj,1]={vj}

$$X[v_j, i] = \bigcup_{v_k v_j \in E(G)} F'[v_k, i-1] \cup \{v_j\}$$

Running time: $2^{k+o(k)} n \log n$

Sets not containing vj

F'[vj,i]=REDUCE(X[vj,i])

Output size: $2^{k+o(k)} \log n$

Running time: $|X[v_j, i]| \, 2^k \log n = 2^{k+o(k)} n 2^k \log n$

Dynamic Programming for k-Path with Reps

F'[vj,1]={vj}

$$X[v_j, i] = \bigcup_{v_k v_j \in E(G)} F'[v_k, i-1] \cup \{v_j\}$$

Running time: $2^{k+o(k)} n \log n$

Sets not containing vj

F'[vj,i]=REDUCE(X[vj,i])

Output size: $2^{k+o(k)} \log n$

Running time: $|X[v_j, i]| \, 2^k \log n = 2^{k+o(k)} n 2^k \log n$

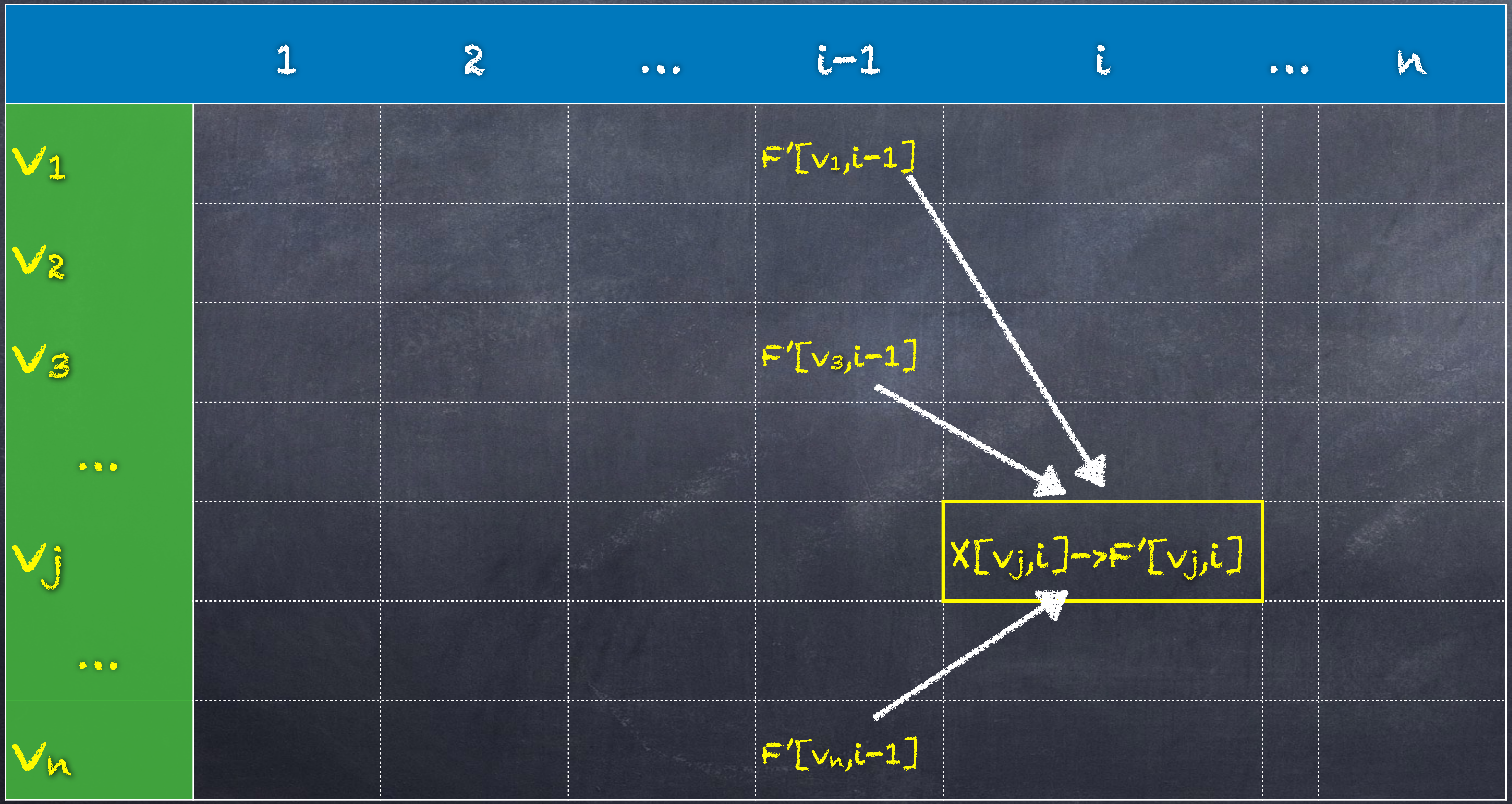
Total running time: $4^{k+o(k)} n^{O(1)}$

Dynamic Programming for k-Path with Reps

Correctness:

We have to show that F'[vj,i] (k-i)-represents F[vj,i]

assuming for each v_k, F'[v_k,i-1] (k-i+1)-represents F[v_k,i-1]



Dynamic Programming for k-Path with Reps Correctness:

Proof:

By def., $F'[v_j, i]$ $(k-i)$-represents $X[v_j, i]$

We show that $X[v_j, i]$ $(k-i)$-reps $F[v_j, i]$

$\Longrightarrow$ $F'[v_j, i]$ $(k-i)$-reps. $F[v_j, i]$

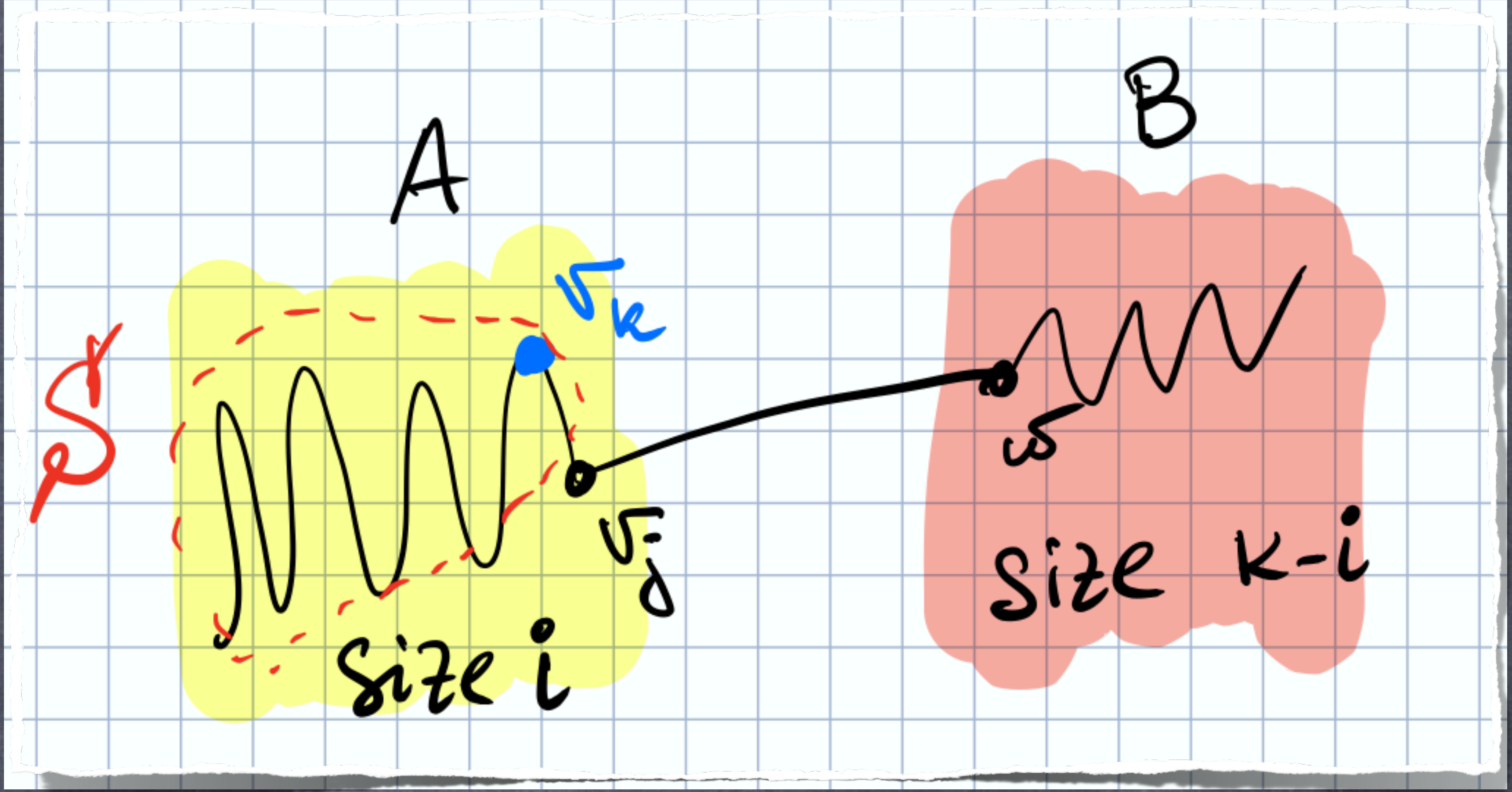
Dynamic Programming for k-Path with Reps Correctness:

Proof:

By def., $F'[v_j, i]$ $(k-i)$-represents $X[v_j, i]$

We show that $X[v_j, i]$ $(k-i)$-reps $F[v_j, i]$

$\Rightarrow F'[v_j, i]$ $(k-i)$-reps. $F[v_j, i]$



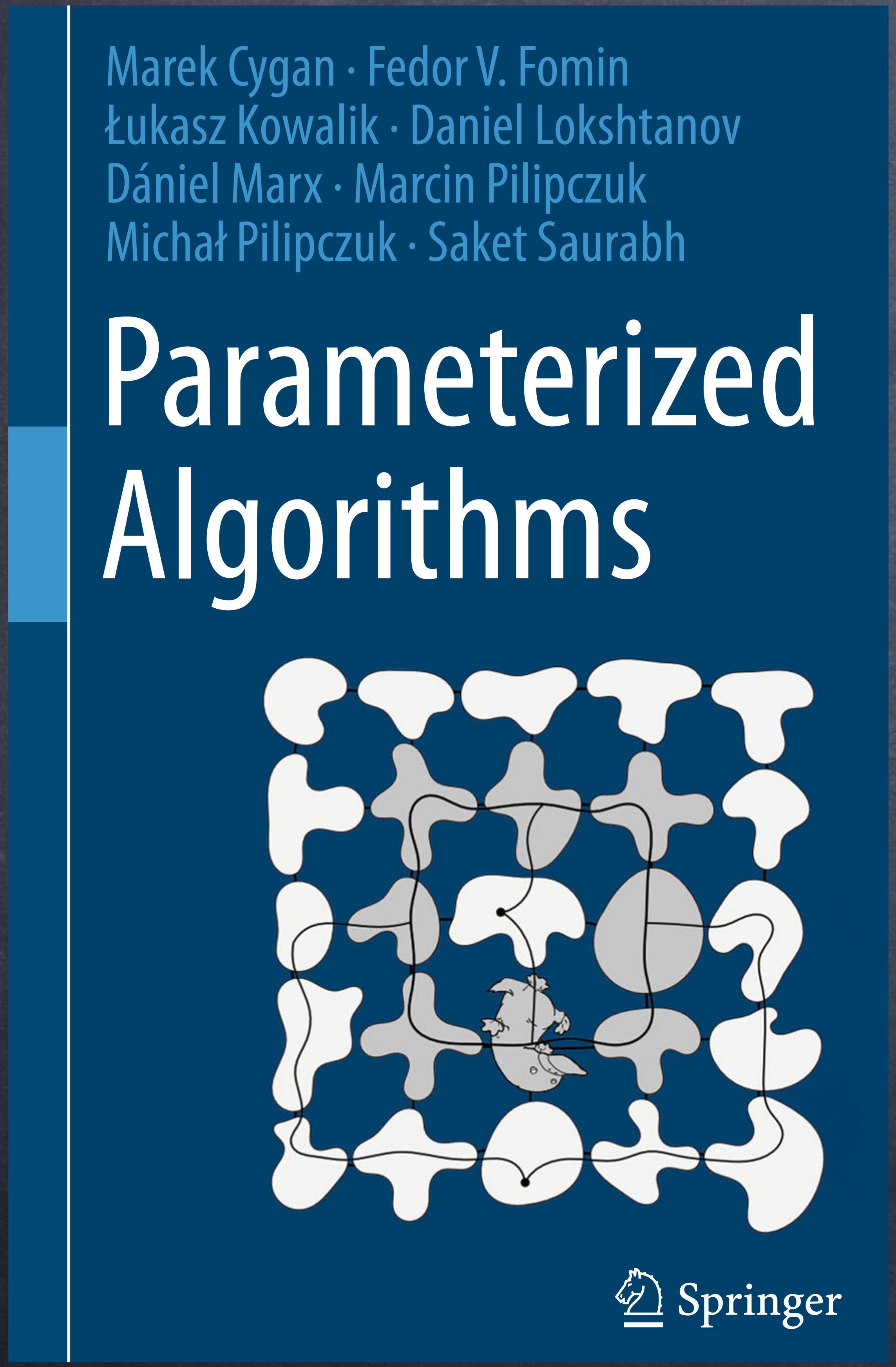
$\exists B$, $|B| = k-i$ and $\exists A \in F[v_j, i]$ s.t. $A \cap B = \emptyset$

Then $\exists v_k$ and $P \in F[v_k, i-1]$ s.t. $P \cap (B \cup \{v_j\}) = \emptyset$

$\Rightarrow$ $\exists S \in F'[v_k, i-1]$ s.t. $S \cap (B \cup \{v_j\}) = \emptyset$.

$S \cup \{v_j\} \in X[v_j, i-1]$ and $(S \cup \{v_j\}) \cap B = \emptyset$ ∎

Further reading



Chapter 12.3

Efficient Computation of Representative Families with Applications in Parameterized and Exact Algorithms

FEDOR V. FOMIN and DANIEL LOKSHTANOV, University of Bergen, Norway
FAHAD PANOLAN, Institute of Mathematical Sciences, India
SAKET SAURABH, Institute of Mathematical Sciences, India, and University of Bergen, Norway

Let $M = (E, \mathcal{I})$ be a matroid and let $\mathcal{S} = \{S_1, \dots, S_t\}$ be a family of subsets of E of size p. A subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is *q-representative* for $\mathcal{S}$ if for every set $Y \subseteq E$ of size at most q, if there is a set $X \in \mathcal{S}$ disjoint from Y with $X \cup Y \in \mathcal{I}$, then there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ disjoint from Y with $\widehat{X} \cup Y \in \mathcal{I}$. By the classic result of Bollobás, in a uniform matroid, every family of sets of size p has a q-representative family with at most $\binom{p+q}{p}$ sets. In his famous "two families theorem" from 1977, Lovász proved that the same bound also holds for any matroid representable over a field $\mathbb{F}$. We give an efficient construction of a q-representative family of size at most $\binom{p+q}{p}$ in time bounded by a polynomial in $\binom{p+q}{p}$, t, and the time required for field operations.

We demonstrate how the efficient construction of representative families can be a powerful tool for designing single-exponential parameterized and exact exponential time algorithms. The applications of our approach include the following:

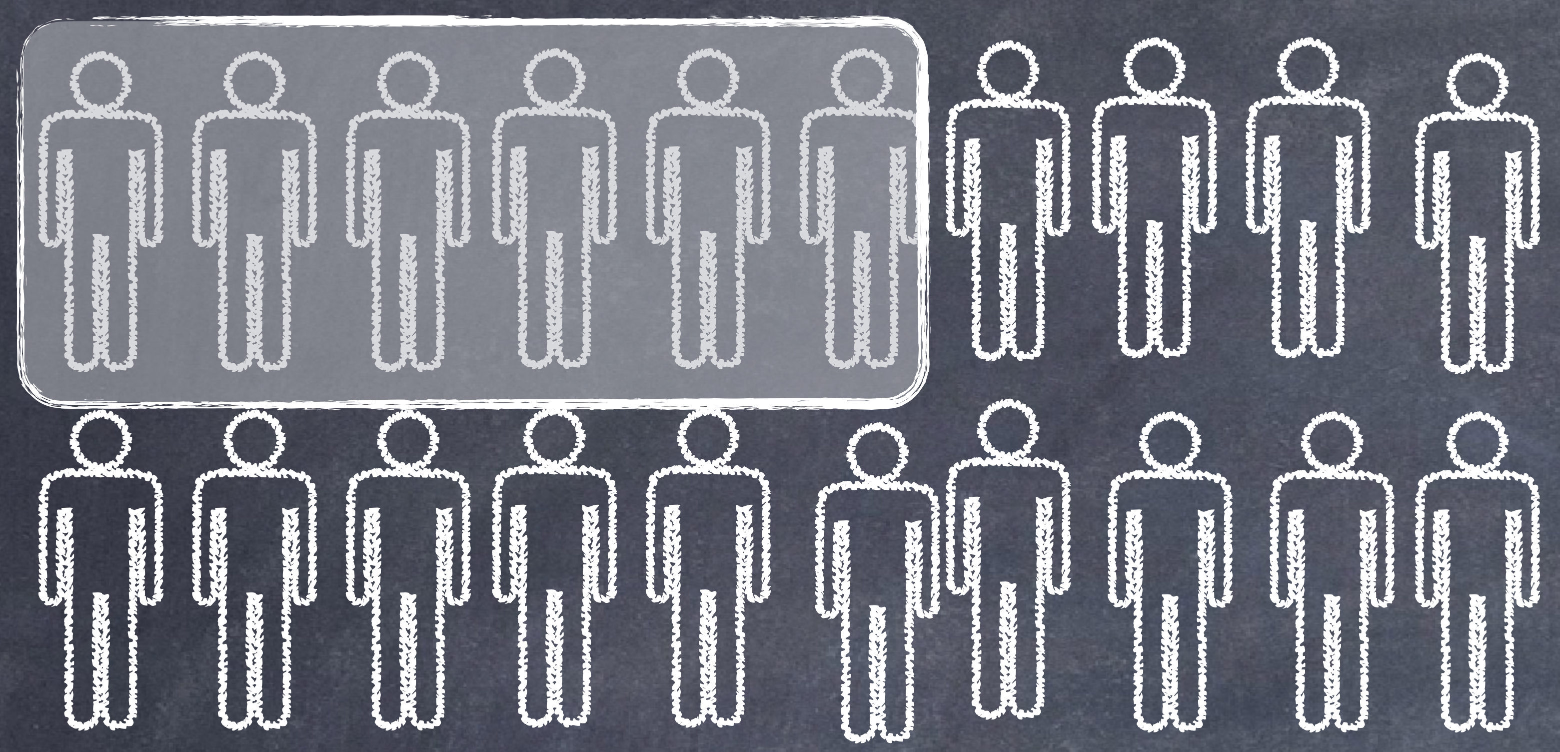
—In the Long Directed Cycle problem, the input is a directed n-vertex graph G and the positive integer k. The task is to find a directed cycle of length at least k in G, if such a cycle exists. As a consequence of our $6.75^{k+o(k)}n^{O(1)}$ time algorithm, we have that a directed cycle of length at least $\log n$, if such a cycle exists, can be found in polynomial time.
—In the Minimum Equivalent Graph (MEG) problem, we are seeking a spanning subdigraph D' of a given n-vertex digraph D with as few arcs as possible in which the reachability relation is the same as in the original digraph D.
—We provide an alternative proof of the recent results for algorithms on graphs of bounded treewidth showing that many "connectivity" problems such as Hamiltonian Cycle or Steiner Tree can be solved in time $2^{O(t)}n$ on n-vertex graphs of treewidth at most t.

For the special case of uniform matroids on n elements, we give a faster algorithm to compute a representative family. We use this algorithm to provide the fastest known deterministic parameterized algorithms for k-Path, k-Tree, and, more generally, k-Subgraph Isomorphism, where the k-vertex pattern graph is of constant treewidth.

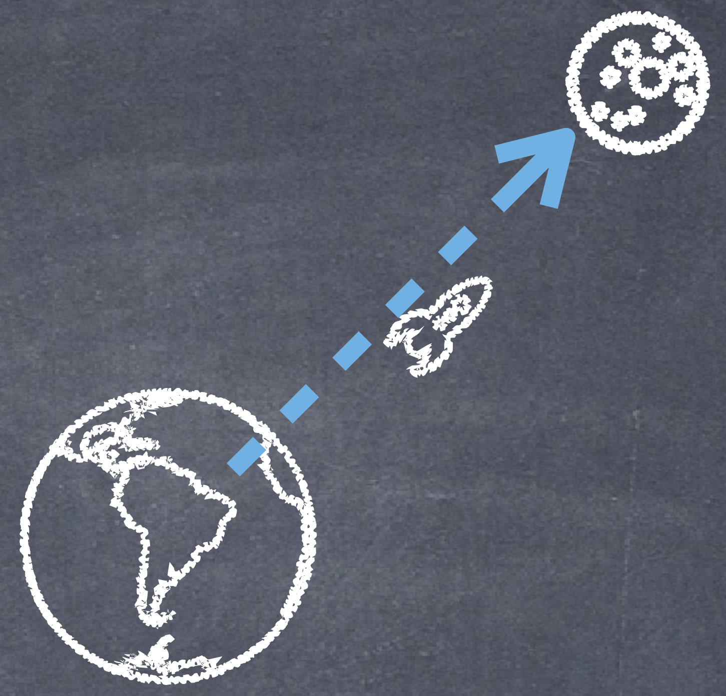
A generalization of Bollobas' Lemma for matroids

- and not only for the sake of generality...

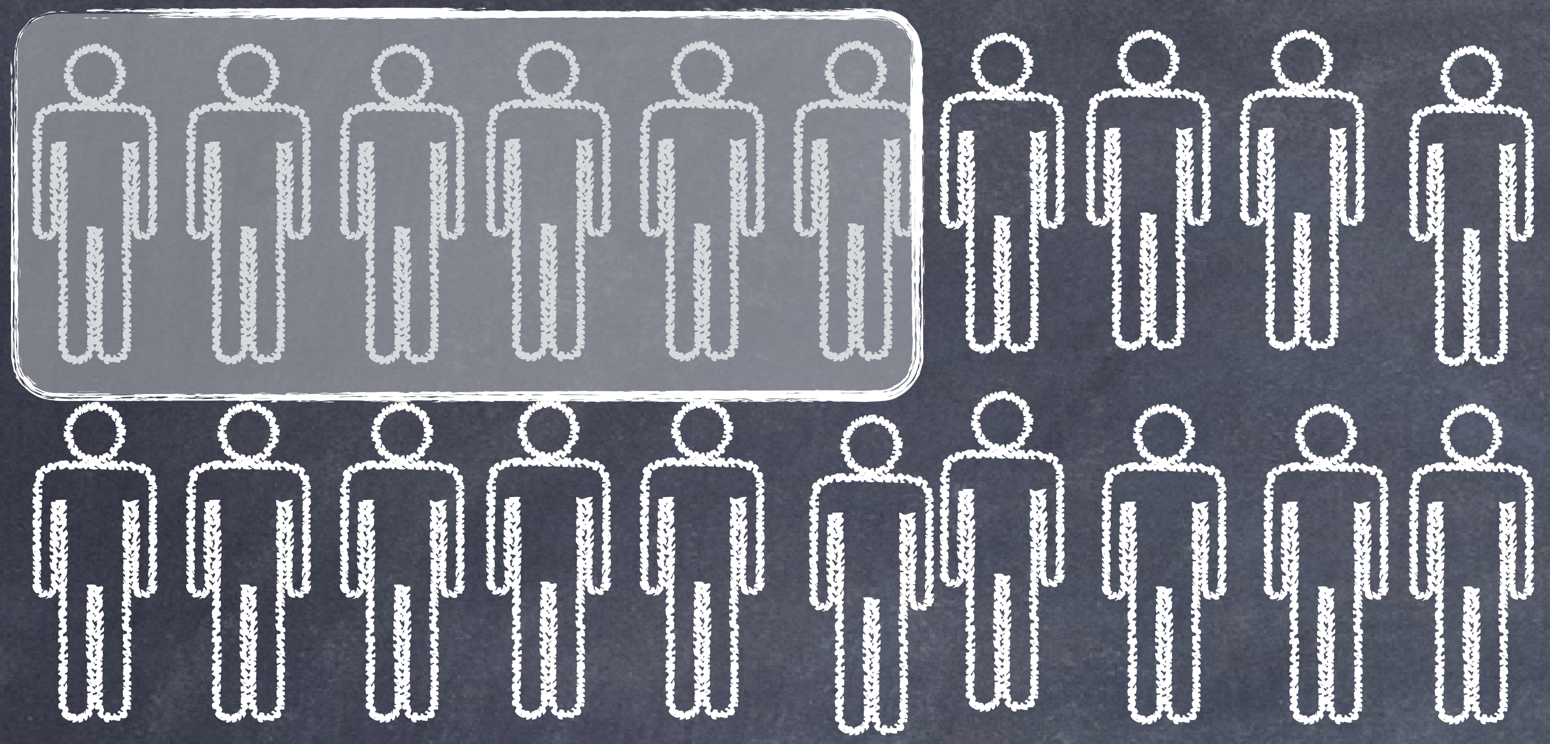
ML Cosmonaut selection

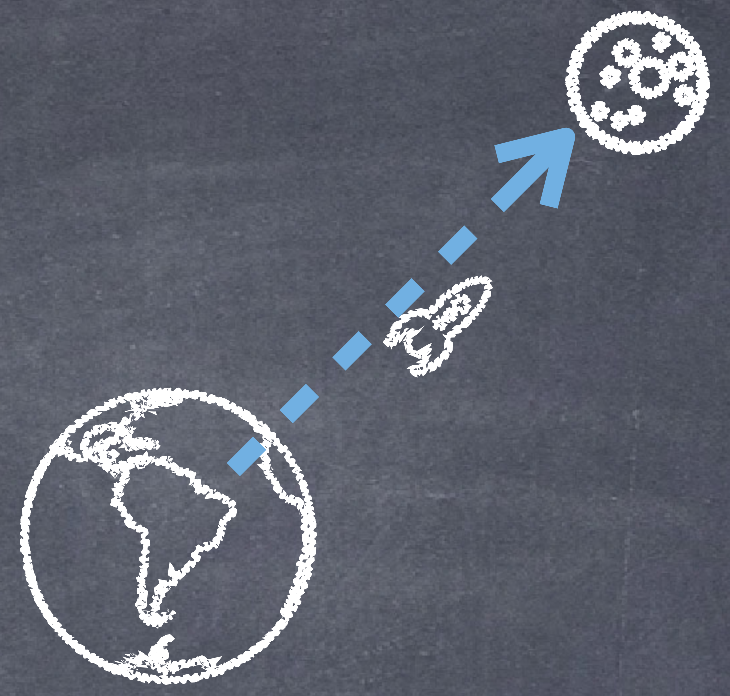


Every cosmonaut has a vector of features
(common friends, favourite movies, etc.)



ML Cosmonaut selection

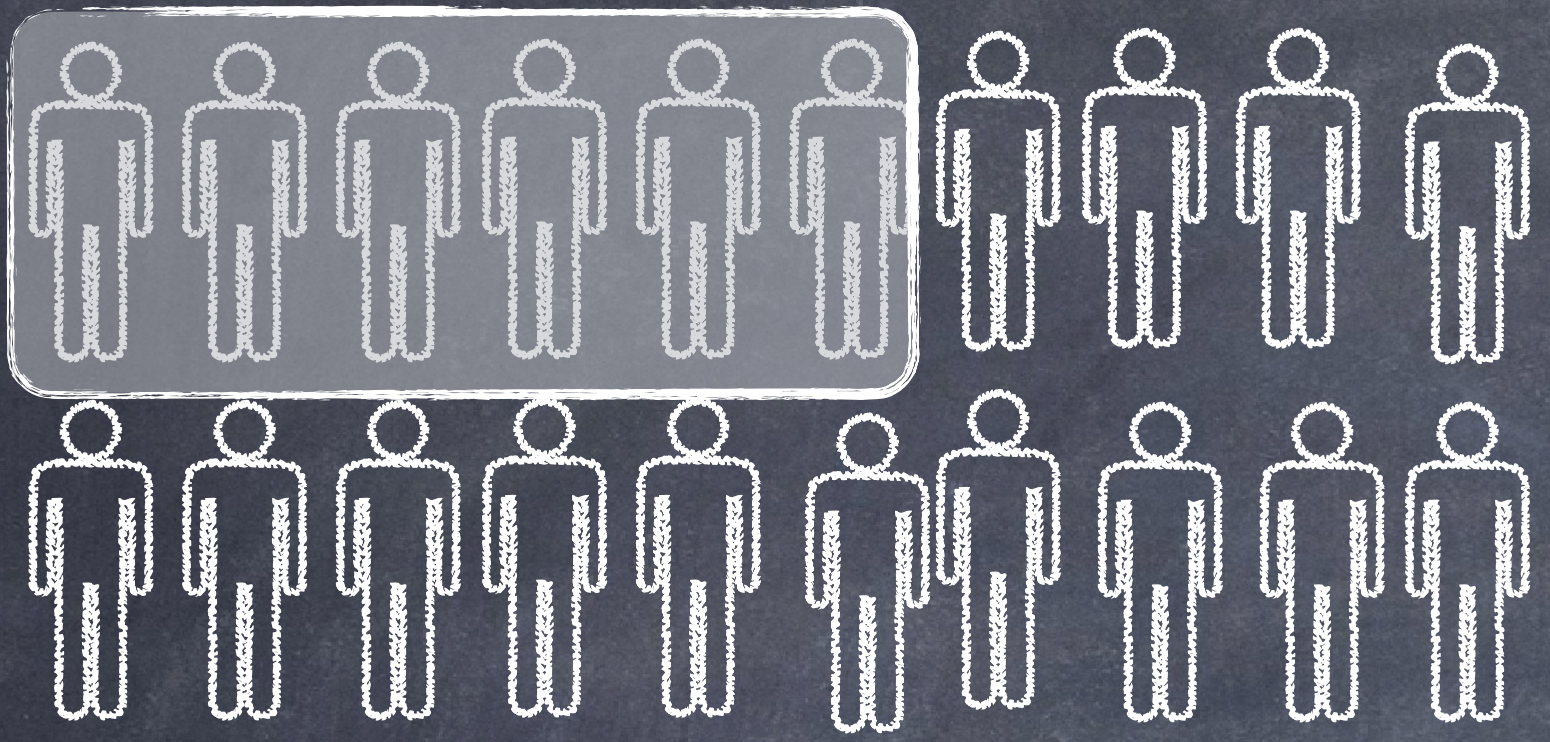


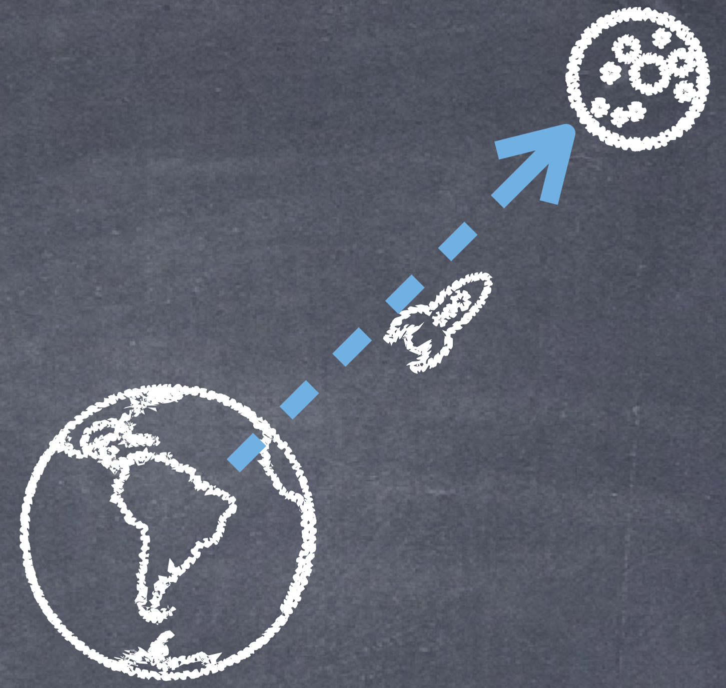


Every cosmonaut has a vector of features
(common friends, favourite movies, etc.)

If 3 cosmonauts will be ill, will at least
one team survive?

ML Cosmonaut selection



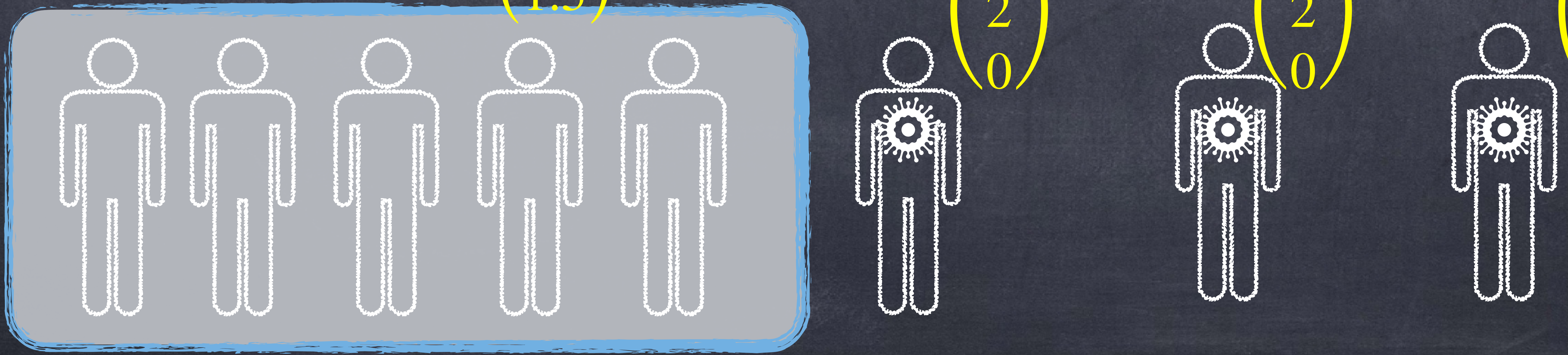


Every cosmonaut has a vector of features (common friends, favourite movies, etc.)

If 3 cosmonauts will be ill, will at least one team survive?

Survive: Means they do not intersect and their features are independent

$$\begin{pmatrix}0\\2\\0\end{pmatrix} \begin{pmatrix}2\\1\\0\end{pmatrix} \begin{pmatrix}2\\-2\\0.3\end{pmatrix} \begin{pmatrix}1.5\\2\\1.5\end{pmatrix} \begin{pmatrix}3\\2\\0\end{pmatrix} \quad \begin{pmatrix}2\\2\\0\end{pmatrix} \quad \begin{pmatrix}1\\2\\0\end{pmatrix} \quad \begin{pmatrix}3\\2\\0.5\end{pmatrix}$$



Matroids

A family F of sets over a finite universe U is a matroid if it satisfies the following three matroid axioms:

- $\emptyset \in \mathcal{F}$,
- if $A \in \mathcal{F}$ and $B \subseteq A$ then $B \in \mathcal{F}$,
- if $A \in \mathcal{F}$ and $B \in \mathcal{F}$ and $|A| < |B|$ then there is an element $b \in B \setminus A$ such that $(A \cup \{b\}) \in \mathcal{F}$.

Matroids

A family F of sets over a finite universe U is a matroid if it satisfies the following three matroid axioms:

- $\emptyset \in \mathcal{F}$,
- if $A \in \mathcal{F}$ and $B \subseteq A$ then $B \in \mathcal{F}$,
- if $A \in \mathcal{F}$ and $B \in \mathcal{F}$ and $|A| < |B|$ then there is an element $b \in B \setminus A$ such that $(A \cup \{b\}) \in \mathcal{F}$.

U edges of matroid
F independent sets of matroid
Maximal independent set - basis
Size of basis - rank

Matroids

A family F of sets over a finite universe U is a **matroid** if it satisfies the following three matroid axioms:

- $\emptyset \in \mathcal{F}$,
- if $A \in \mathcal{F}$ and $B \subseteq A$ then $B \in \mathcal{F}$,
- if $A \in \mathcal{F}$ and $B \in \mathcal{F}$ and $|A| < |B|$ then there is an element $b \in B \setminus A$ such that $(A \cup \{b\}) \in \mathcal{F}$.

U **edges** of matroid
F **independent sets** of matroid
Maximal independent set - **basis**
Size of basis - **rank**

If you never saw matroid!!!

Think of U as vectors (over some field) and F as linearly independent sets of vectors.

Important Matroids

Take some field

In our applications we often use GF(s)

Important Matroids

Take some field

In our applications we often use GF(s)

Addition

+	O	I	A	B
O	O	I	A	B
I	I	O	B	A
A	A	B	O	I
B	B	A	I	O

Multiplication

·	O	I	A	B
O	O	O	O	O
I	O	I	A	B
A	O	A	B	I
B	O	B	I	A

Example GF(4), source Wikipedia

Important Matroids

Take some field

In our applications we often use GF(s)

For edge $e \in U$ -> vector v_e over some field

Addition						Multiplication				
+	0	1	A	B		·	0	1	A	B
0	0	1	A	B		0	0	0	0	0
1	1	0	B	A		1	0	1	A	B
A	A	B	0	1		A	0	A	B	1
B	B	A	1	0		B	0	B	1	A

Example GF(4), source Wikipedia

Important Matroids

Take some field

In our applications we often use GF(s)

Addition

+	O	I	A	B
O	O	I	A	B
I	I	O	B	A
A	A	B	O	I
B	B	A	I	O

Multiplication

·	O	I	A	B
O	O	O	O	O
I	O	I	A	B
A	O	A	B	I
B	O	B	I	A

Example GF(4), source Wikipedia

For edge $e \in U$ -> vector v_e over some field

Declare $S \subseteq U$ independent if and only if $\{v_e : e \in S\}$ is linearly independent

Important Matroids

Take some field

In our applications we often use GF(s)

Addition					Multiplication				
+	O	I	A	B	·	O	I	A	B
O	O	I	A	B	O	O	O	O	O
I	I	O	B	A	I	O	I	A	B
A	A	B	O	I	A	O	A	B	I
B	B	A	I	O	B	O	B	I	A

Example GF(4), source Wikipedia

For edge $e \in U$ -> vector v_e over some field

Why this is a matroid?

Declare $S \subseteq U$ independent if and only if $\{v_e : e \in S\}$ is linearly independent

Important Matroids. Linear matroid

Declare $S \subseteq U$ independent if and only if $\{v_e \ : \ e \in S\}$ is linearly independent

$$M = \begin{array}{c|ccccc|} & e_1 & e_2 & e_3 & \cdots & e_m \\ \hline 1 & * & * & * & \cdots & * \\ 2 & * & * & * & \cdots & * \\ 3 & * & & & & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ell & * & * & * & \cdots & * \end{array}_{\ell \times m}$$

Elements of F

Important Matroids. Linear matroid

Declare $S \subseteq U$ independent if and only if $\{v_e \,:\, e \in S\}$ is linearly independent

$$
M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ \ell \end{array}
\begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & \cdots & e_m \end{array} \\
\begin{bmatrix} * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ * & & \text{Elements of } F & & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & * & \cdots & * \end{bmatrix}_{\ell \times m} \end{array}
$$

M represents matroid over field F

Important Matroids. Linear matroid

Declare $S \subseteq U$ independent if and only if $\{v_e \; : \; e \in S\}$ is linearly independent

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ \ell \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & \cdots & e_m \end{array} \\ \left[\begin{array}{ccccc} * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ * & & \text{Elements of F} & & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & * & \cdots & * \end{array} \right]_{\ell \times m} \end{array}$$

M represents matroid over field F

Matroid is linear or representable

Uniform Matroid

A pair $\mathscr{M} = (U, \mathscr{F})$ over an n-element ground set U, is a uniform matroid if the family of independent sets is given by

$$\mathscr{F} = \{A \subseteq U \ : \ |A| \leq k\}$$

where k is some constant. This matroid is also denoted as $U_{n,k}$

Example: $U = \{1,2,3,4\}$ and $k = 2$

$\mathscr{F} = \{\{\emptyset\}, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$

Uniform Matroid

A pair $\mathscr{M} = (U, \mathscr{F})$ over an n-element ground set U, is a uniform matroid if the family of independent sets is given by

$$\mathscr{F} = \{A \subseteq U \ : \ |A| \leq k\}$$

Why this is a matroid?

where k is some constant. This matroid is also denoted as $U_{n,k}$

Example: U = {1,2,3,4} and k = 2

$\mathscr{F} = \{\{\emptyset\}, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$

Uniform Matroid is representable

A pair $\mathscr{M} = (U, \mathscr{F})$ over an n-element ground set U, is a uniform matroid if the family of independent sets is given by

$\mathscr{F} = \{A \subseteq U \ : \ |A| \leq k\}$

where k is some ≈. This matroid is also denoted as $U_{n,k}$

$U_{n,k}$ is representable over GF(p), p>n.

Uniform Matroid is representable

A pair $\mathscr{M} = (U, \mathscr{F})$ over an n-element ground set U, is a uniform matroid if the family of independent sets is given by
$\mathscr{F} = \{A \subseteq U \ : \ |A| \leq k\}$
where k is some $\mathbb{N}$. This matroid is also denoted as $U_{n,k}$

$U_{n,k}$ is representable over GF(p), p>n.

$U = \{e_1, \ldots, e_n\}$, for each e_i assign non-zero field element α_i and vector $\begin{pmatrix} 1 \\ \alpha_1 \\ \alpha_1^2 \\ \vdots \\ \alpha_1^{k-1} \end{pmatrix}$

Uniform Matroid is representable

A pair $\mathscr{M} = (U, \mathscr{F})$ over an n-element ground set U, is a uniform matroid if the family of independent sets is given by

$\mathscr{F} = \{A \subseteq U \ : \ |A| \leq k\}$

where k is some ≈. This matroid is also denoted as $U_{n,k}$

$U_{n,k}$ is representable over GF(p), p>n.

$U = \{e_1, \ldots, e_n\}$, for each e_i assign non-zero field element α_i and vector $\begin{pmatrix} 1 \\ \alpha_1 \\ \alpha_1^2 \\ \vdots \\ \alpha_1^{k-1} \end{pmatrix}$

$$M = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \cdots & \alpha_n^{k-1} \end{pmatrix}$$

Uniform Matroid is representable

A pair $\mathscr{M} = (U, \mathscr{F})$ over an n-element ground set U, is a uniform matroid if the family of independent sets is given by

$\mathscr{F} = \{A \subseteq U \ : \ |A| \leq k\}$

where k is some ≈. This matroid is also denoted as $U_{n,k}$

$U_{n,k}$ is representable over GF(p), p>n.

$U = \{e_1, \ldots, e_n\}$, for each e_i assign non-zero field element α_i and vector $\begin{pmatrix} 1 \\ \alpha_1 \\ \alpha_1^2 \\ \vdots \\ \alpha_1^{k-1} \end{pmatrix}$

$$M = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \cdots & \alpha_n^{k-1} \end{pmatrix}$$

k+1 columns are linearly dependent

For set A of k columns, the determinant of the Vandermonde matrix M_A is $\prod_{i<j,\ e_i,e_j \in A} \alpha_j - \alpha_i \neq 0$

Graphic Matroid

For a graph G, a graphic matroid is defined as $\mathscr{M} = (U, \mathscr{F})$ where U = E(G) (edges of G are elements of the matroid)

$\mathscr{F} = \{A \subseteq U \ : \ A \text{ is a forest}\}$

Graphic Matroid

For a graph G, a graphic matroid is defined as $\mathscr{M} = (U, \mathscr{F})$ where U = E(G) (edges of G are elements of the matroid)

$\mathscr{F} = \{A \subseteq U \ : \ A \text{ is a forest}\}$

Why this is a matroid?

Graphic Matroid

For a graph G, a graphic matroid is defined as $\mathscr{M} = (U, \mathscr{F})$ where U = E(G) (edges of G are elements of the matroid)

$\mathscr{F} = \{A \subseteq U \ : \ A \text{ is a forest}\}$

Why this is a matroid?

Is it a representable matroid?

Representative Sets and Matroids

Reps for matroids

Let M be a matroid. Set A fits B if $A \cap B = \emptyset$ and $A \cup B$ is independent.

Let M be a uniform matroid of rank $a+b$. a-set A fits b-set B iff $A \cap B = \emptyset$

Representative Sets and Matroids

Reps for matroids

Let M be a matroid. Set A fits B if $A \cap B = \emptyset$ and $A \cup B$ is independent.

Reps for matroids

Let M be a matroid and F be a family of a-sets in M. A subfamily $F' \subseteq F$ b-represents F if for every B of size b such that there exists an $A \in F$ that fits B, there exists an $A' \in F'$ that also fits B.

Representative Sets and Matroids

Reps for matroids

Let M be a matroid. Set A fits B if $A \cap B = \emptyset$ and $A \cup B$ is independent.

Reps for matroids

Let M be a matroid and F be a family of a-sets in M. A subfamily $F' \subseteq F$ b-represents F if for every B of size b such that there exists an $A \in F$ that fits B, there exists an $A' \in F'$ that also fits B.

Reps for sets

Let F be a family of a-sets, a subfamily $F' \subseteq F$ b-represents F if for every B of size b such that there exists an $A \in F$ with $A \cap B = \emptyset$ there exists an $A' \in F'$ with $A' \cap B = \emptyset$.

FLATS IN MATROIDS AND GEOMETRIC GRAPHS

L. LOVÁSZ

Bolyai Institute, József Attila University,

Szeged, Hungary

1. INTRODUCTION

This paper was intended to deal with the covering problems in graphs. It has turned out, however, that their study becomes much simpler if a more general structure, which we shall call geometric graph, is considered. Some problems on the covering number of graphs can be translated then to Helly-type problems concerning flats in matroids. The solution of these Helly-type problems (which is complete for the representable matroids only) has required some operations on matroids which generalize the Kronecker product of matrices or versions of it. Analogous Helly-type problems on flats in

Theorem There is an algorithm that, given a matrix M over a field GF(s), representing a matroid $\mathcal{M} = (U, \mathcal{F})$ of rank k, an a-family $\mathcal{A}$ of independent sets in $\mathcal{M}$, and an integer b such that a+b=k, computes a b-representative family $\mathcal{A}'$ of $\mathcal{A}$ of size at most $\binom{a+b}{a}$ using at most $O(|\mathcal{A}|(\binom{a+b}{b}b^{\omega} + \binom{a+b}{b}^{\omega-1}))$ operations in GF(s).

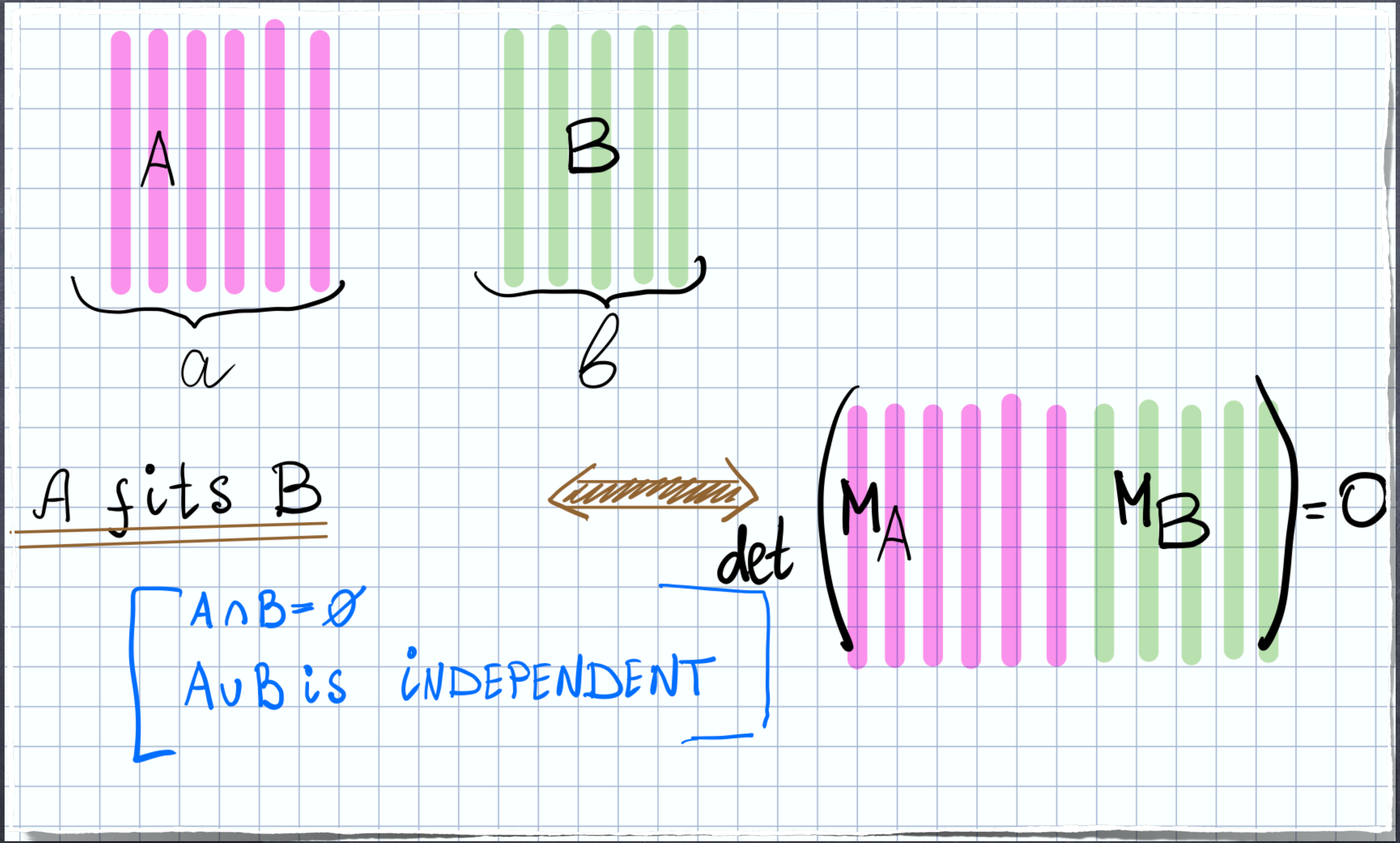
$\omega = 2.73...$ matrix multiplication exponent

Theorem There is an algorithm that, given a matrix M over a field GF(s), representing a matroid $\mathscr{M} = (U, \mathscr{F})$ of rank k, an a-family $\mathscr{A}$ of independent sets in $\mathscr{M}$, and an integer b such that a+b=k, computes a b-representative family $\mathscr{A}'$ of $\mathscr{A}$ of size at most $\binom{a+b}{a}$ using at most $O(|\mathscr{A}|(\binom{a+b}{b}b^{\omega} + \binom{a+b}{b}^{\omega-1}))$ operations in GF(s).

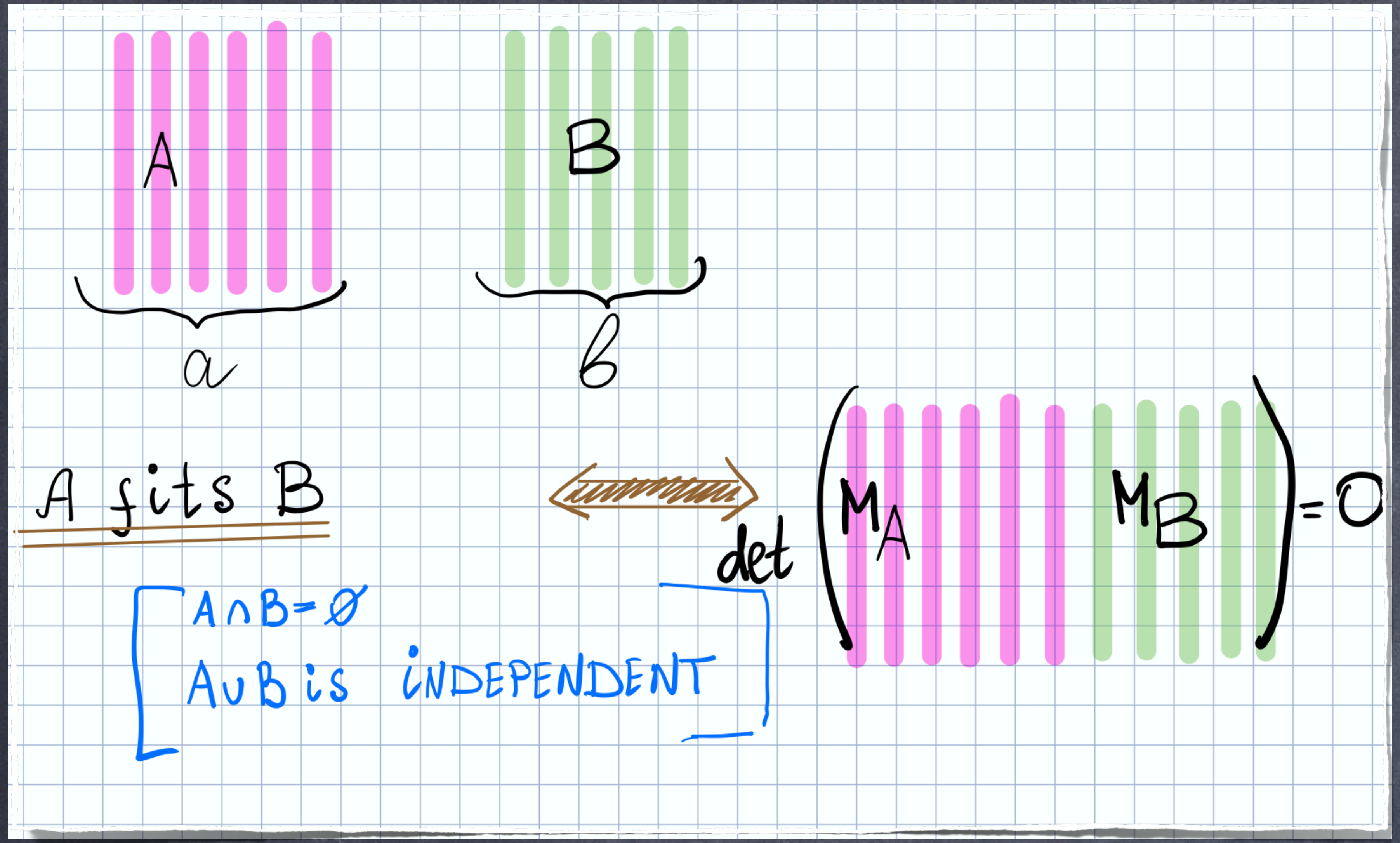
Polynomial in $|\mathscr{A}|$

$\omega = 2.73...$ matrix multiplication exponent

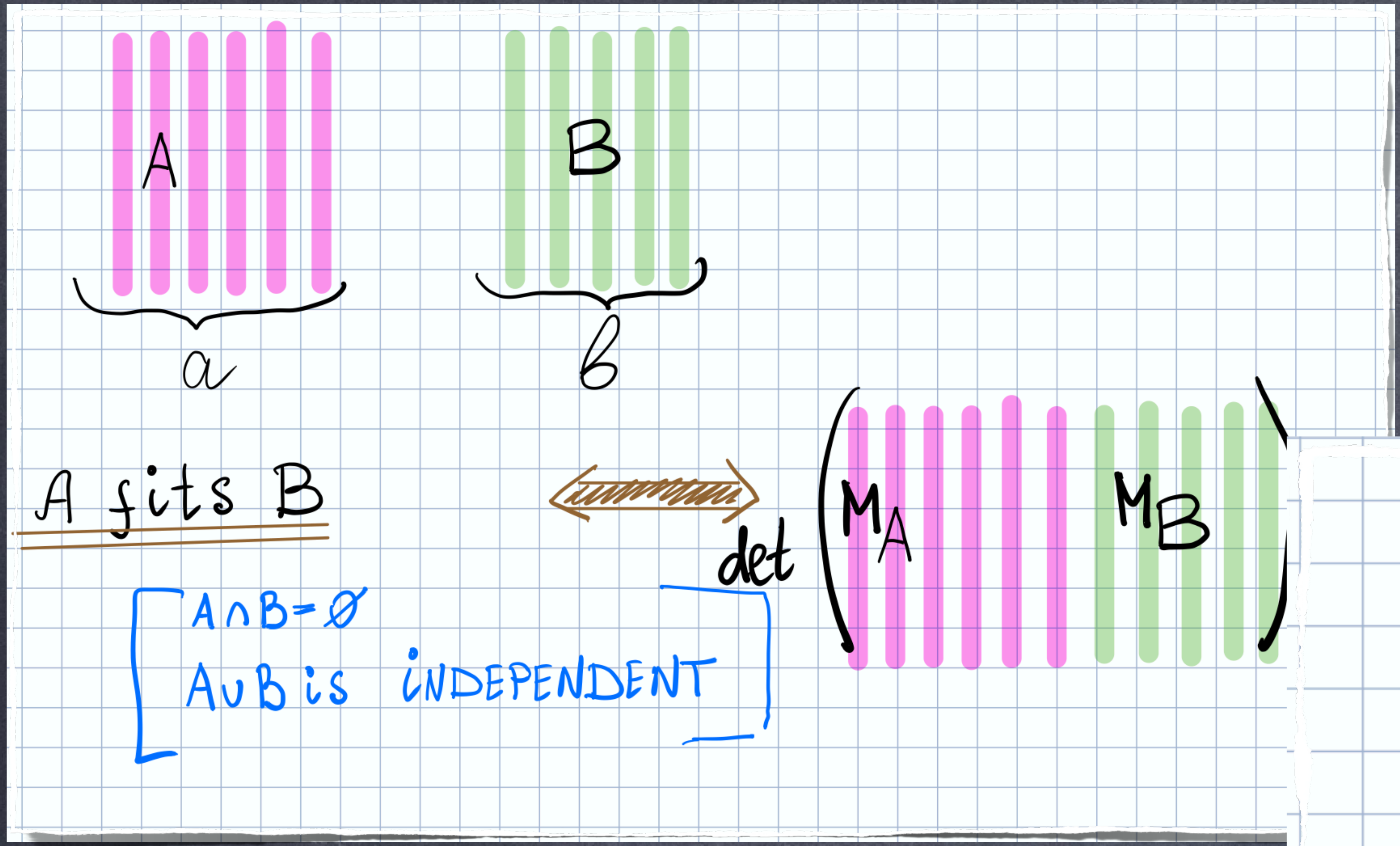
Proof: Exterior (Grassmann) algebra under the carpet



Proof: Exterior (Grassmann) algebra under the carpet



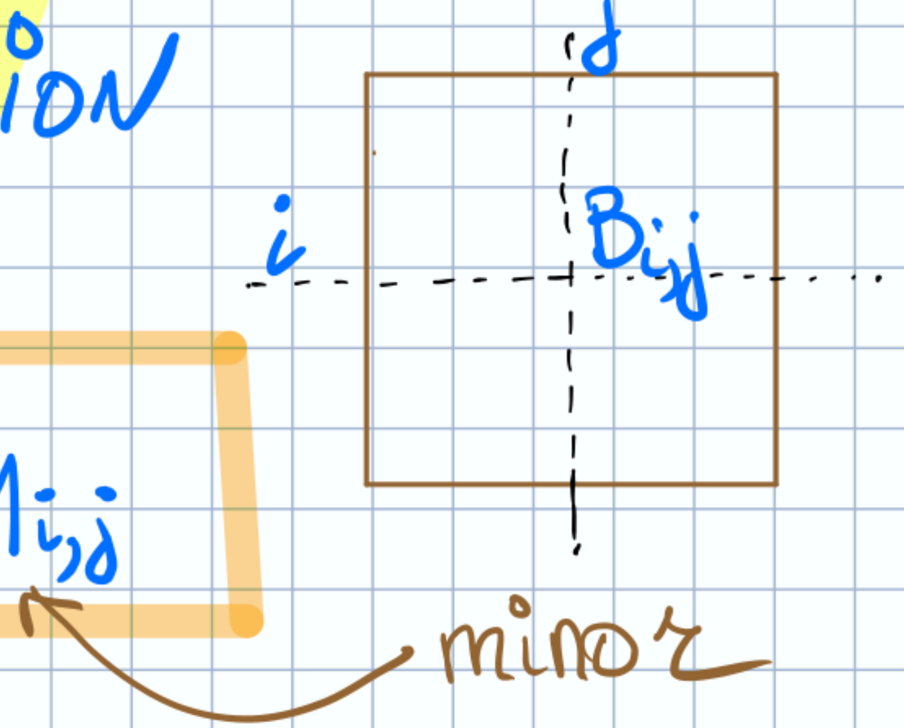
Proof: Exterior (Grassmann) algebra under the carpet



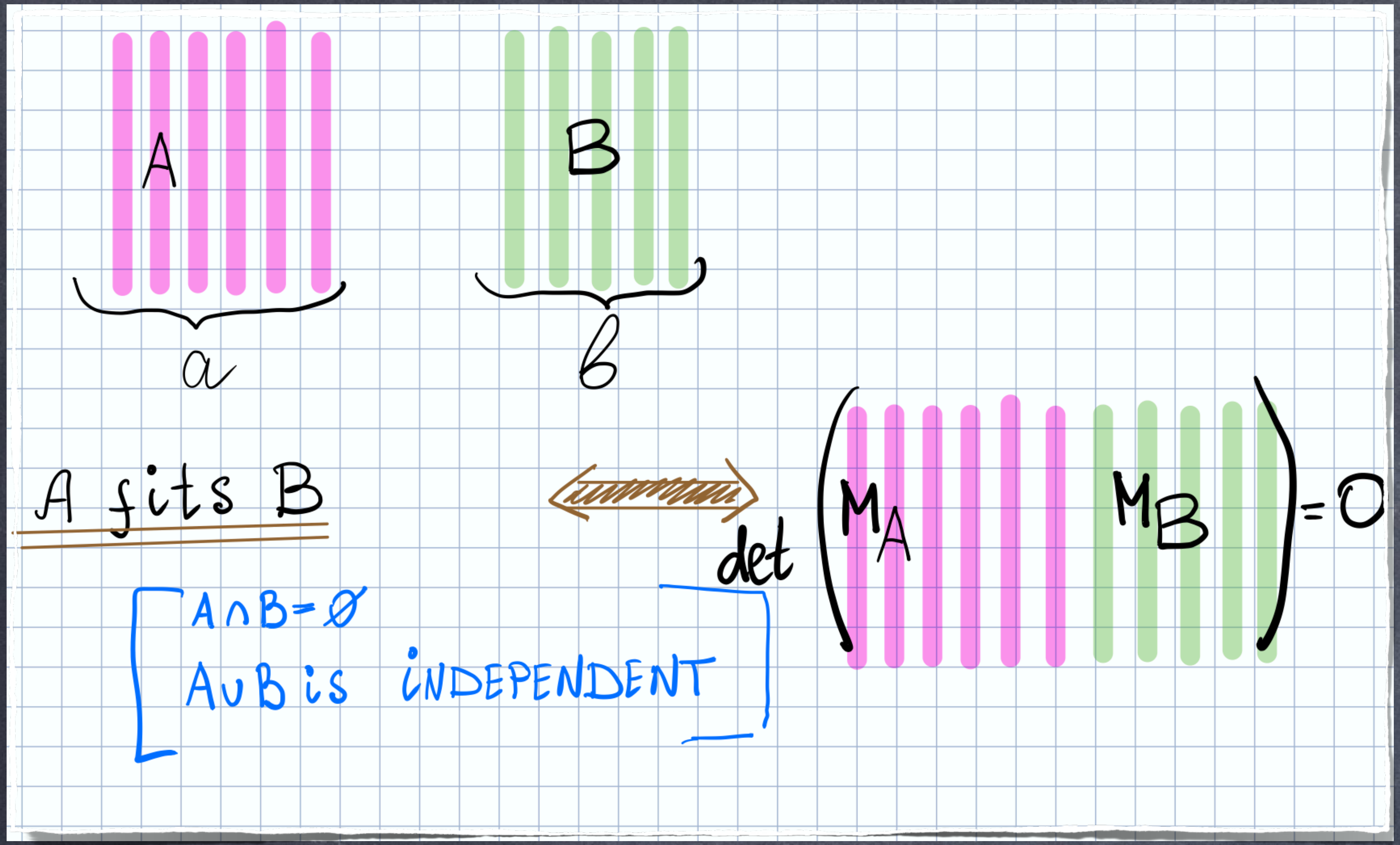
Laplace expansion

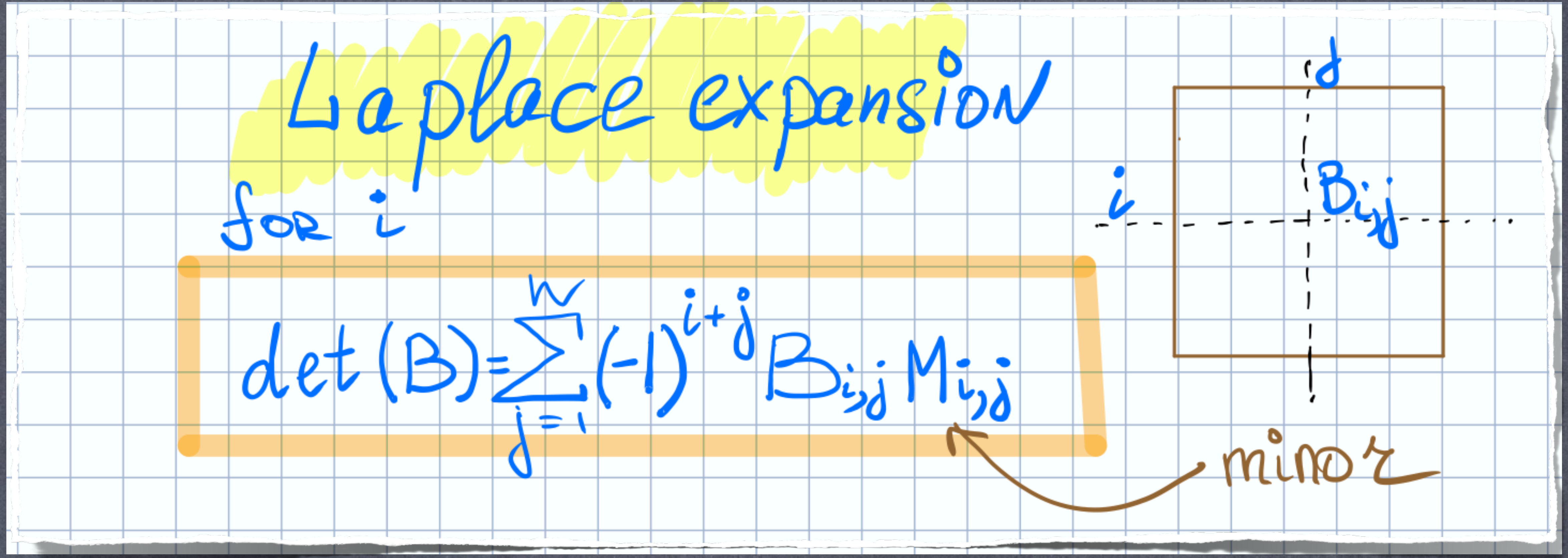
for i

$$\det(B)=\sum_{j=1}^{n}(-1)^{i+j}B_{i,j}M_{i,j}$$

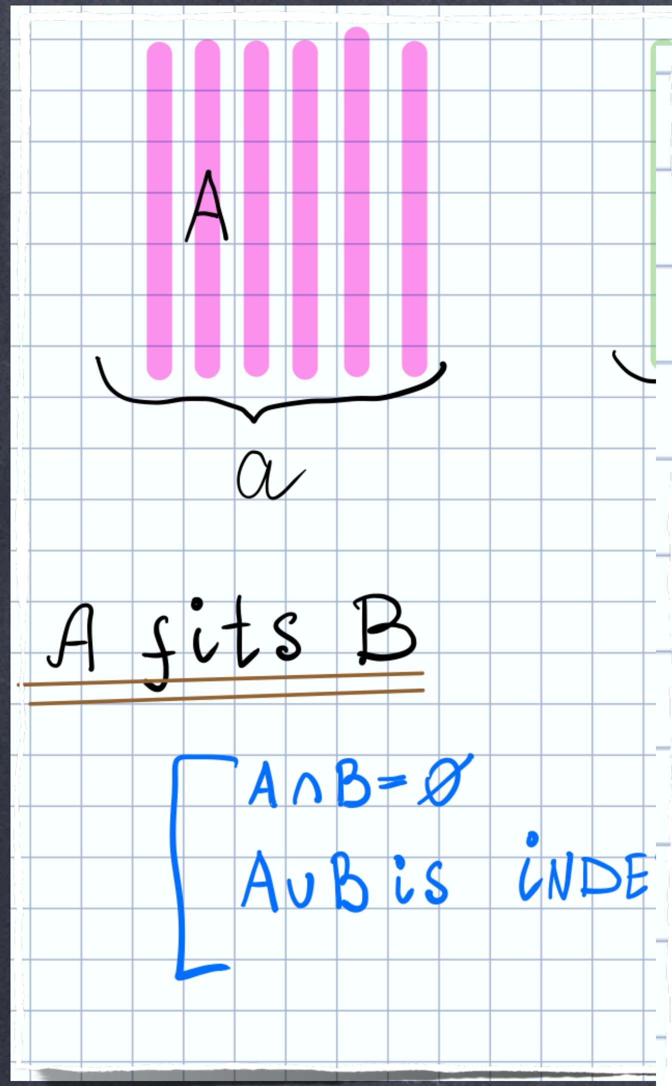


Proof: Exterior (Grassmann) algebra under the carpet

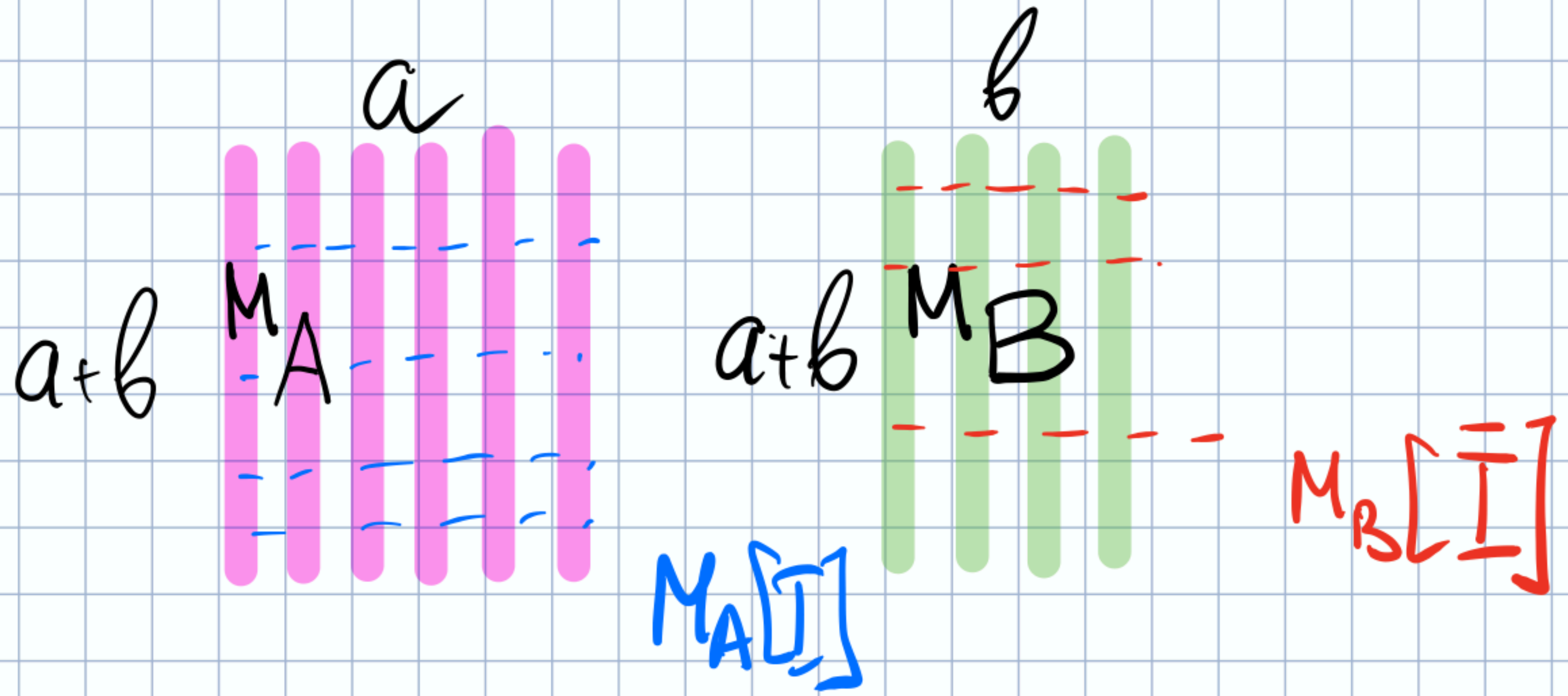


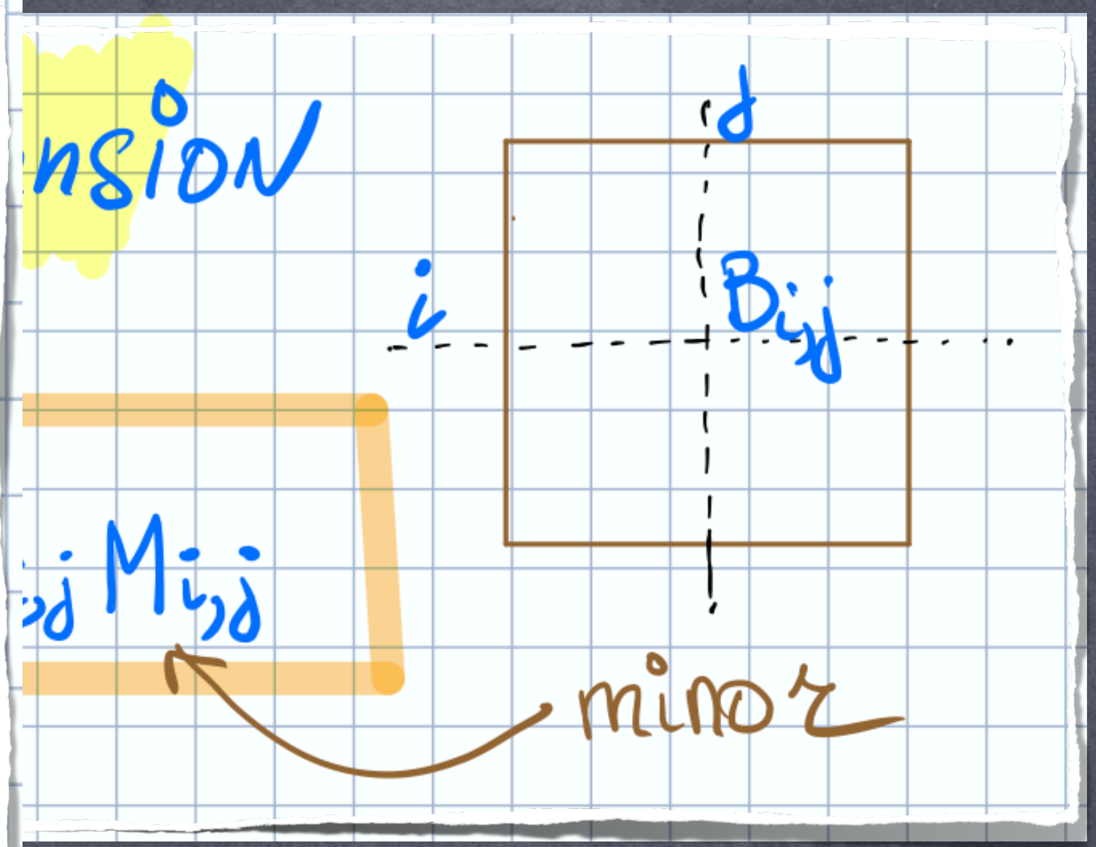


Proof: Exterior (Grassmann) algebra under the carpet



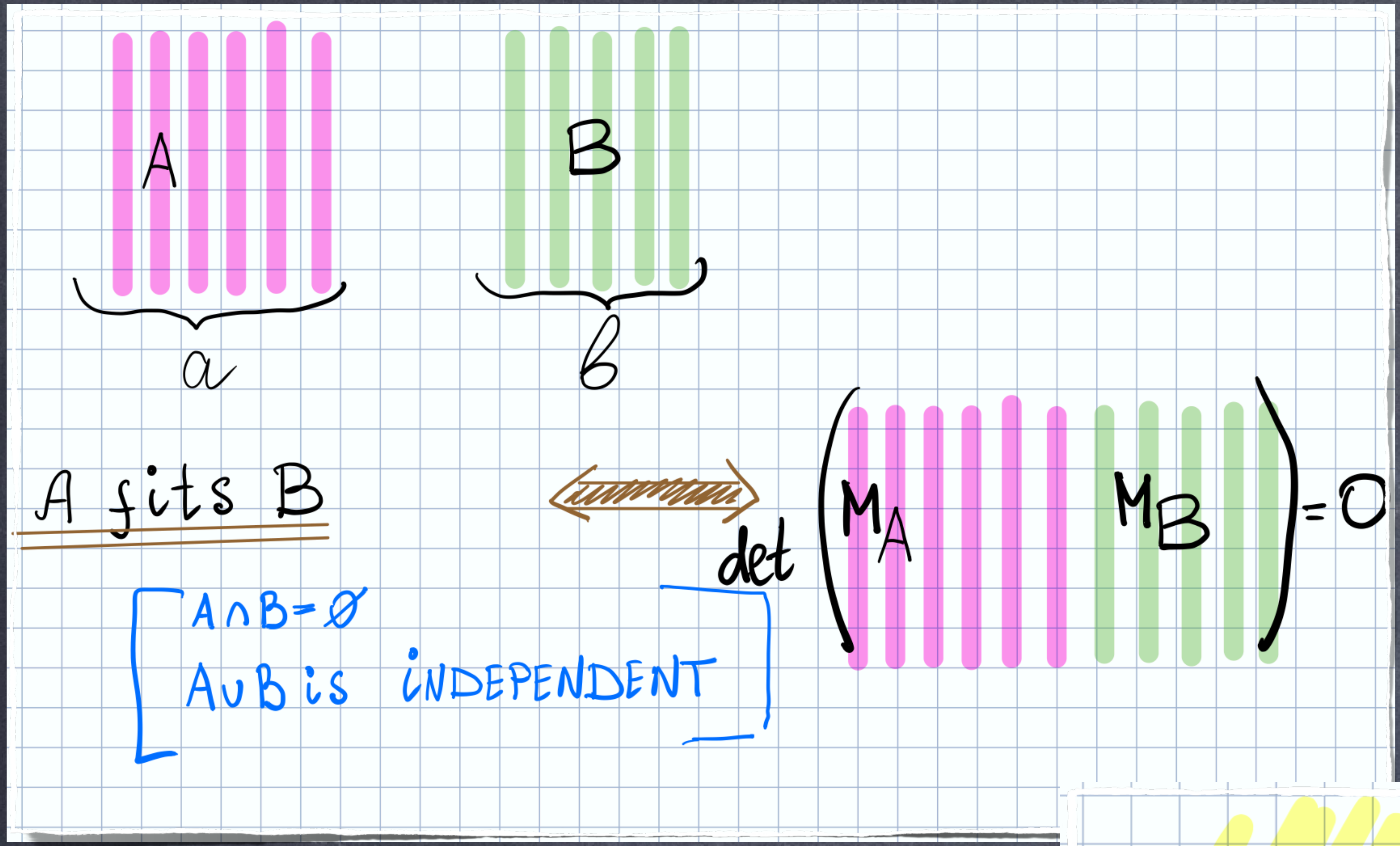
Generalized Laplace expansion

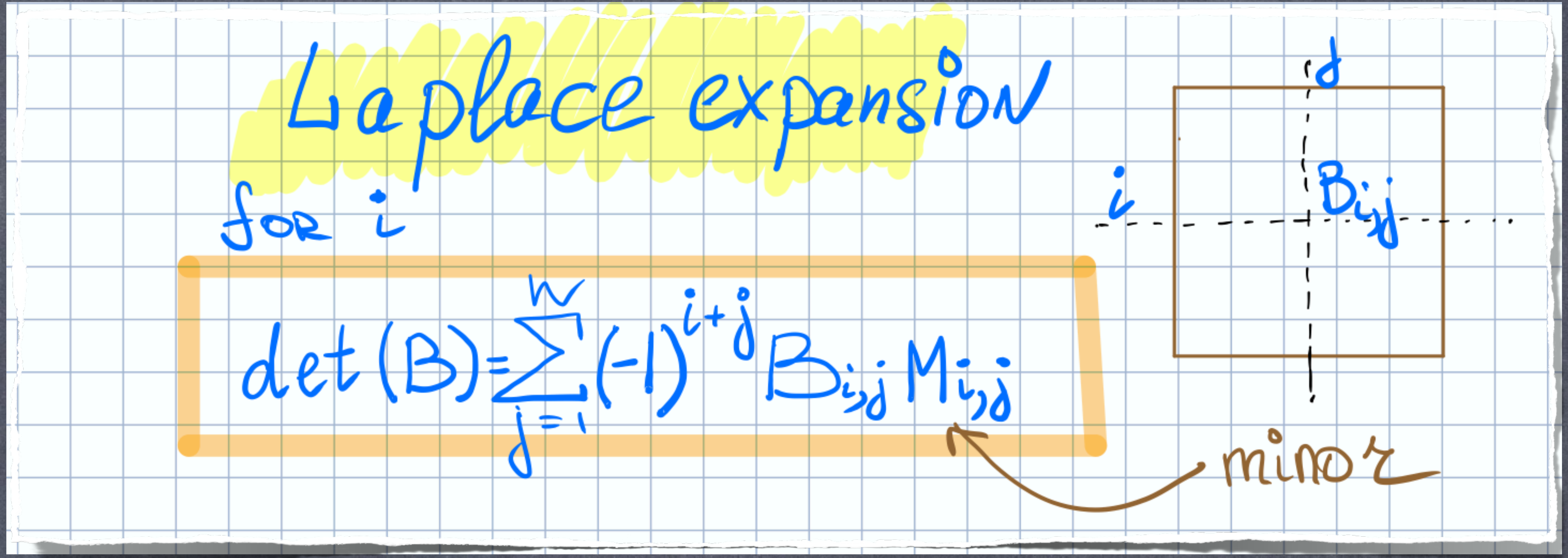


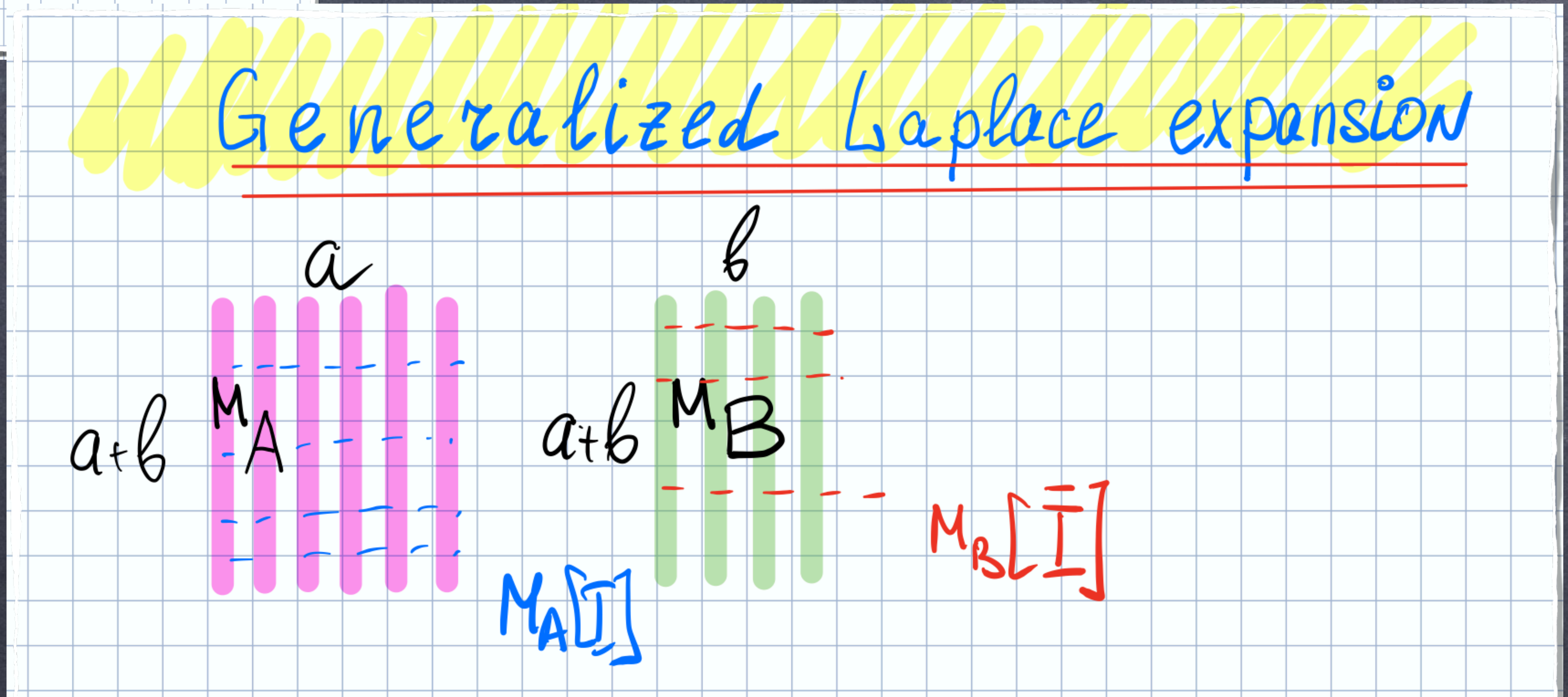


$$\det([M_A|M_B]) = (-1)^{\lceil a/2\rceil} \sum_{I\subseteq[a+b],|I|=a} (-1)^{\Sigma I} \det(M_A[I])\det(M_B[\bar{I}]).$$

Proof: Exterior (Grassmann) algebra under the carpet







$$\det([M_A|M_B]) = (-1)^{\lceil a/2 \rceil} \sum_{I\subseteq[a+b],|I|=a} (-1)^{\sum I} \det(M_A[I]) \det(M_B[\bar{I}]).$$

Proof

$I_1, \ldots, I_\ell$: all a-subsets of $\{1, 2, \ldots, a+b\}$

$$\ell = \binom{a+b}{a}$$

Proof

$I_1, \dots, I_\ell$: all a-subsets of $\{1, 2, \dots, a+b\}$

$\ell = \binom{a+b}{a}$

For a-set A define $v^A \in \mathbb{R}^\ell$

$$v^A = \begin{pmatrix} \vdots \\ \cdots \; v_i^A \\ \vdots \end{pmatrix} \qquad v_i^A = \det\left(M_A[I_i]\right)$$

Proof

$I_1, \dots, I_\ell$: all a-subsets of $\{1, 2, \dots, a+b\}$

$$\ell = \binom{a+b}{a}$$

For a-set A define $v^A \in \mathbb{R}^\ell$

$$v^A = \begin{pmatrix} \vdots \\ v_i^A \\ \vdots \end{pmatrix} \quad v_i^A = \det\left(M_A[I_i]\right)$$

Proof

$I_1, \dots, I_\ell$: all a-subsets of $\{1,2,\dots,a+b\}$

$\ell = \binom{a+b}{a}$

For a-set A define $v^A \in \mathbb{R}^\ell$

$v^A = \begin{pmatrix} \vdots \end{pmatrix}$ $v_i^A = \det(M_A[I_i])$

$\overline{I}_i = \{1, \dots, a+b\} \setminus I_i$

For b-set B define $\overline{v}^B \in \mathbb{R}^\ell$

$\overline{v}^B = \begin{pmatrix} \vdots \end{pmatrix}_i$ $v_i^B = \det(M_B[\overline{I}_i])$

Proof

$I_1, \dots, I_\ell$: all a-subsets of $\{1, 2, \dots, a+b\}$

$$\ell = \binom{a+b}{a}$$

For a-set A define $v^A \in \mathbb{R}^\ell$

$$v^A = \begin{pmatrix} \vdots \\ v_i^A \\ \vdots \end{pmatrix} \qquad v_i^A = \det\left(M_A[I_i]\right)$$

$$\bar{I}_i = \{1, \dots, a+b\} \setminus I_i$$

For b-set B define $\bar{v}^B \in \mathbb{R}^\ell$

$$\bar{v}^B = \begin{pmatrix} \vdots \\ \bar{v}_i^B \\ \vdots \end{pmatrix} \qquad \bar{v}_i^B = \det\left(M_B[\bar{I}_i]\right)$$

Proof

$I_1, \dots, I_\ell$: all a-subsets of $\{1,2,\dots,a+b\}$

$$\ell = \binom{a+b}{a}$$

$$\sigma_i = (-1)^{\sum I_i}$$

For a-set A define $v^A \in \mathbb{R}^\ell$

$$v^A = \begin{pmatrix} \vdots \\ v_i^A \\ \vdots \end{pmatrix}, \quad v_i^A = \det\left(M_A[I_i]\right)$$

$\bar{I}_i = \{1,\dots,a+b\} \setminus I_i$

For b-set B define $\bar{v}^B \in \mathbb{R}^\ell$

$$\bar{v}^B = \begin{pmatrix} \vdots \\ \bar{v}_i^B \\ \vdots \end{pmatrix}, \quad \bar{v}_i^B = \det\left(M_B[\bar{I}_i]\right)$$

Proof

$I_1, \ldots, I_\ell$: all a-subsets of $\{1, 2, \ldots, a+b\}$

$\ell = \binom{a+b}{a}$

For a-set A define $v^A \in \mathbb{R}^\ell$

$v^A = \begin{pmatrix} \vdots \\ \cdot \\ \vdots \end{pmatrix}_i$, $v_i^A = \det(M_A[I_i])$

$\bar{I}_i = \{1, \ldots, a+b\} \setminus I_i$

For b-set B define $\bar{v}^B \in \mathbb{R}^\ell$

$\bar{v}^B = \begin{pmatrix} \vdots \\ \cdot \\ \vdots \end{pmatrix}_i$, $v_i^B = \det(M_B[\bar{I}_i])$

$\sigma_i = (-1)^{\sum I_i}$

Proof

$I_1, \dots, I_\ell$: all a-subsets of $\{1,2,\dots,a+b\}$

$$\ell = \binom{a+b}{a}$$

For a-set A define $v^A \in \mathbb{R}^\ell$

$$v^A = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix} \quad i \qquad v_i^A = \det\left(M_A[I_i]\right)$$

$$\bar{I}_i = \{1, \dots, a+b\} \setminus I_i$$

For b-set B define $\bar{v}^B \in \mathbb{R}^\ell$

$$\bar{v}^B = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix} \quad i \qquad \bar{v}_i^B = \det\left(M_B[\bar{I}_i]\right)$$

$$\sigma_i = (-1)^{\sum I_i}$$

Observation

$$A \text{ fits } B \iff \det(M_A | M_B) \neq 0$$

$$\iff \sum_{i=1}^{\ell} \sigma_i v_i^A \cdot \bar{v}_i^B \neq 0 .$$

Proof

$I_1, \dots, I_\ell$: all a-subsets of $\{1, 2, \dots, a+b\}$

$\ell = \binom{a+b}{a}$

For a-set A define $v^A \in \mathbb{R}^\ell$

$v^A = \begin{pmatrix} \vdots \\ \cdots \\ \vdots \end{pmatrix} i$, $v_i^A = \det(M_A[I_i])$

$\bar{I}_i = \{1, \dots, a+b\} \setminus I_i$

For b-set B define $\bar{v}^B \in \mathbb{R}^\ell$

$\bar{v}^B = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix} i$, $\bar{v}_i^B = \det(M_B[\bar{I}_i])$

Observation

A fits B $\iff$ $\det(M_A | M_B) \neq 0$

$$\sum_{i=1}^{\ell} \sigma_i v_i^A \cdot \bar{v}_i^B \neq 0$$

$$\det([M_A|M_B]) = (-1)^{\lceil a/2 \rceil} \sum_{I \subseteq [a+b], |I| = a} (-1)^{\sum I} \det(M_A[I]) \det(M_B[\bar{I}]).$$

$$\sigma_i = (-1)^{\sum I_i}$$

Proof: Main idea

We know : Matrix M, set A $\Rightarrow$ σ_A

We do not know : Set B $\Rightarrow$ $\overline{\sigma}_B$

What we care : $\sum_{i=1}^{\ell} 6_i \, \sigma_i^A \cdot \overline{\sigma}_i^B \neq \emptyset$

Proof: Main idea

We know : Matrix M, set $A \Rightarrow v_A$

We do not know : Set $B \Rightarrow \bar{v}_B$

What we care : $\sum_{i=1}^{\ell} \sigma_i \, v_i^A \cdot \bar{v}_i^B \neq \emptyset$

IDEA: Just the basis of $\{v^A \mid A \in \mathcal{A}\}$ suffice!

size $|\mathcal{F}|$

size $\binom{a+b}{a}$

Proof: Algorithm

Algorithm

Compute basis $\mathfrak{A}'$ of $\{v^A \mid A \in \mathfrak{A}\}$

a column

Proof: Algorithm

Algorithm

Compute basis $\mathcal{A}'$ of $\{v^A \mid A \in \mathcal{A}\}$

a column

TIME:

- v^A: $\binom{a+b}{a} a^{\omega}$

  \# of entries (pointing to $\binom{a+b}{a}$)

  computing determ. of $a \times a$ matrix (pointing to a^{ω})

- $\{v^A \mid A \in \mathcal{A}\}$: $|\mathcal{A}| \cdot \binom{a+b}{a} \cdot a^{\omega}$

- Basis $\mathcal{A}'$: $\binom{a+b}{a} \Big(\; \Big)$ with $|\mathcal{A}|$ columns

  $n \times m$ matrix in $mn^{\omega-1}$

  $\rightarrow |\mathcal{A}| \cdot \binom{a+b}{a}^{\omega-1}$

Proof: Correctness

Correctness

→ $\mathcal{A}'$ basis of $\binom{a+b}{b} \times |\mathcal{A}|$ matrix

$\Rightarrow |\mathcal{A}'| \leq \binom{a+b}{b}$

Proof: Correctness

Correctness

→ $\mathcal{A}'$ basis of $\binom{a+b}{b} \times |\mathcal{A}|$ matrix

⇒ $|\mathcal{A}'| \le \binom{a+b}{b}$

→ Why $\mathcal{A}'$ b-represents $\mathcal{A}$?

$\exists B$ fits some $A \in \mathcal{A}$

⇕

$$0 \neq \sum_{i=1}^{\ell} 6_i v_i^A \bar{v}_i^B = \sum_{i=1}^{\ell} \sum_{A' \in \mathcal{A}'} 6_i \lambda_{A'} v_i^{A'} \bar{v}_i^B$$

$$= \sum_{A' \in \mathcal{A}'} \lambda_{A'} \sum_{i=1}^{\ell} 6_i v_i^{A'} \bar{v}_i^B$$

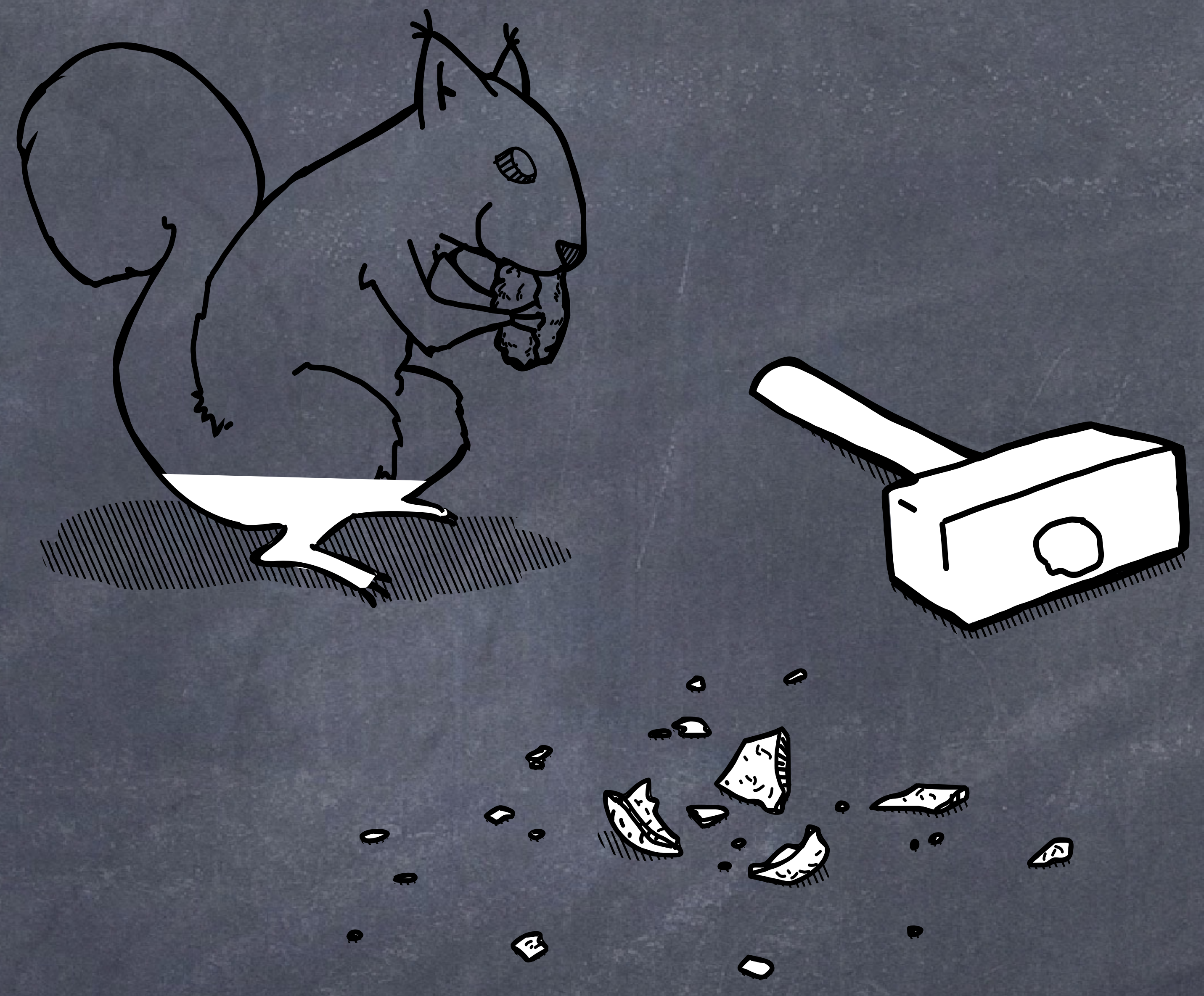
⇙

B fits some $A' \in \mathcal{A}'$ ∎

Summarizing

The algorithm computes a b-representative family $\mathscr{A}'$ of $\mathscr{A}$ of size at most $\binom{a+b}{a}$ using at most $O(|\mathscr{A}|(\binom{a+b}{b}b^{\omega}+\binom{a+b}{b}^{\omega-1}))$ operations.

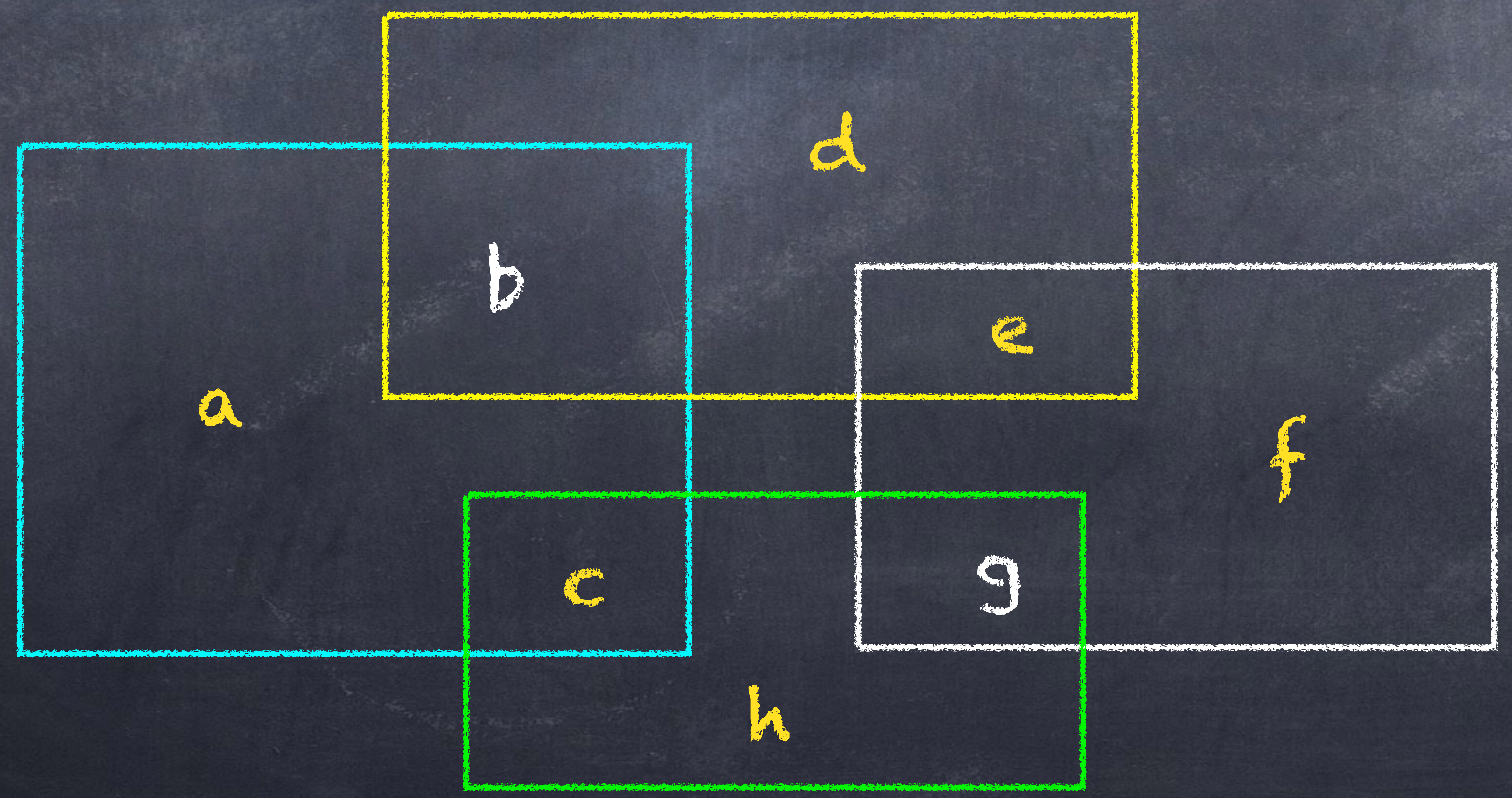
Application. Kernelization



d-Hitting Set

Input: A universe U, a family S of sets of size d over U, integer k

Question: Does there exist a subset X of U of size k that has a nonempty intersection with every member of S?



Polynomial kernel: What we shoot for

A polynomial time algorithm that takes as an input an instance of d-Hitting Set

A universe U, a family S of sets of size d over U, integer k

Polynomial kernel: What we shoot for

A polynomial time algorithm that takes as an input an instance of d-Hitting Set

A universe U, a family S of sets of size d over U, integer k

Outputs an equivalent instance of d-Hitting Set

Polynomial kernel: What we shoot for

A polynomial time algorithm that takes as an input an instance of d-Hitting Set

A universe U, a family S of sets of size d over U, integer k

Outputs an equivalent instance of d-Hitting Set

A universe U', a family S' of sets of size d over U', integer k'

Polynomial kernel: What we shoot for

A polynomial time algorithm that takes as an input an instance of d-Hitting Set

A universe U, a family S of sets of size d over U, integer k

Outputs an equivalent instance of d-Hitting Set

A universe U', a family S' of sets of size d over U', integer k'

Such that the size of the new instance is bounded by a polynomial of k.

Reminder from Lecture 1

2-Hitting Set (Vertex Cover)

Theorem: Every edge k-critical graph has at most $\binom{k+1}{2}$ edges

"Algorithm": If graph is not k-critical, delete an edge

Reminder from Lecture 1

2-Hitting Set (Vertex Cover)

A graph G is edge k-critical, if its vertex cover is k, but for every edge e, the vertex cover of $G \setminus e$ is at most $k-1$.

Theorem: Every edge k-critical graph has at most $\binom{k+1}{2}$ edges

"Algorithm": If graph is not k-critical, delete an edge

d-Hitting Set

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Hitting Set admits a kernel with at most $\binom{k+d}{d}$ sets and at most $\binom{k+d}{d} \cdot d$ elements

Polynomial time algorithm producing an equivalent instance

d-Hitting Set

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Hitting Set admits a kernel with at most $\binom{k+d}{d}$ sets and at most $\binom{k+d}{d} \cdot d$ elements

d-Hitting Set

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Hitting Set admits a kernel with at most $\binom{k+d}{d}$ sets and at most $\binom{k+d}{d} \cdot d$ elements

Algorithm: Compute a k-representative set S' of S, delete all elements not in S'.

d-Hitting Set

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Hitting Set admits a kernel with at most $\binom{k+d}{d}$ sets and at most $\binom{k+d}{d} \cdot d$ elements

Algorithm: Compute a k-representative set S′ of S, delete all elements not in S′.

Compute representation matrix of uniform matroid of rank k+d

d-Hitting Set

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Hitting Set admits a kernel with at most $\binom{k+d}{d}$ sets and at most $\binom{k+d}{d} \cdot d$ elements

Algorithm: Compute a k-representative set S′ of S, delete all elements not in S′.

Compute representation matrix of uniform matroid of rank k+d

$$|S'| \leq \binom{k+d}{d} \text{ and } |U'| \leq \binom{k+d}{d} \cdot d$$

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

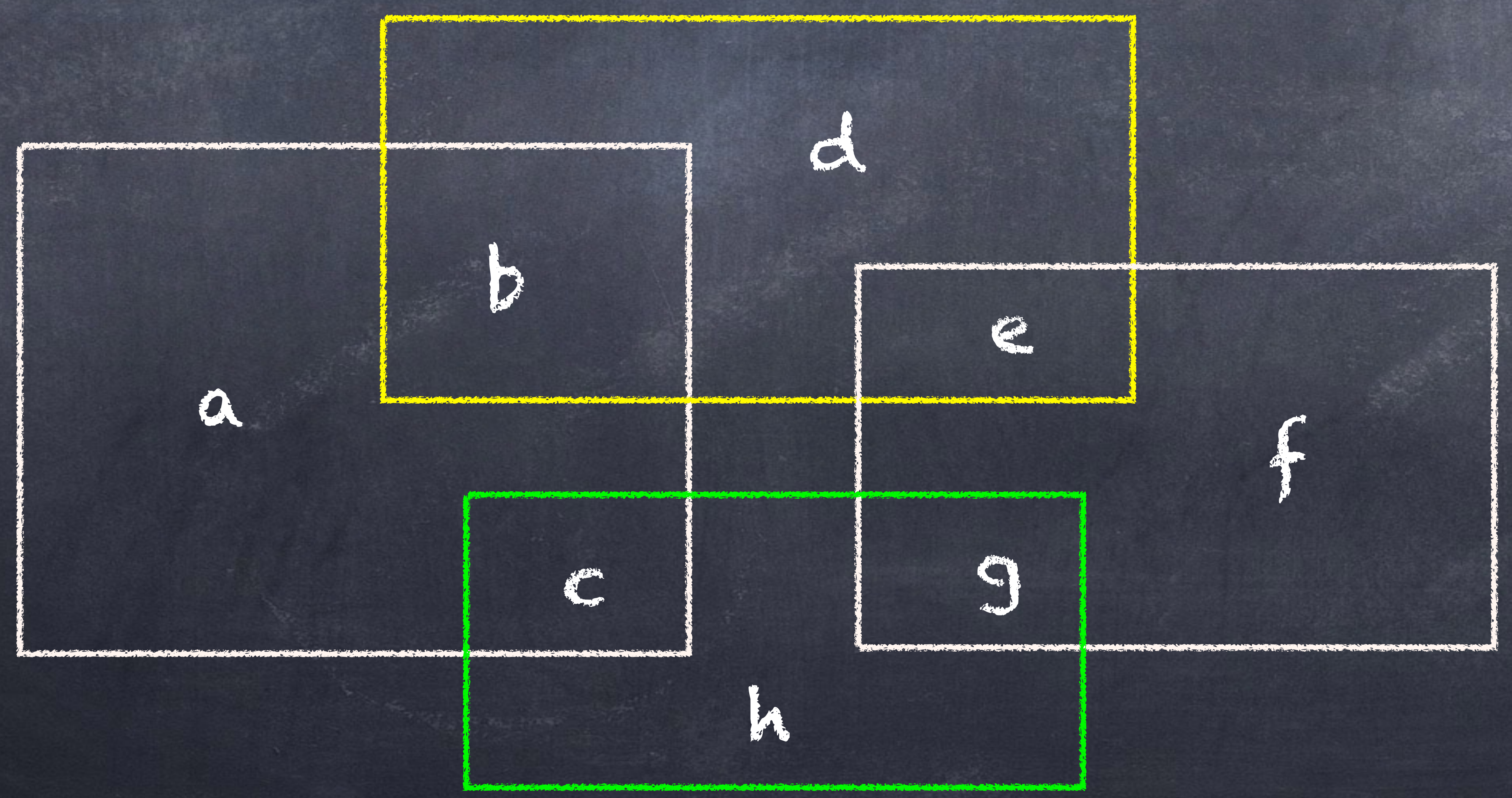
-> If a set B of size k hits every set in S, it also hits every set in S'.

<- If a set B of size k is not a hitting set for S, there is A in S that fits B. Then there is A' in S' that also fits B. Hence B is not a hitting set for S'.

d-Set Packing

Input: A universe U, a family S of sets of size at most d over U, integer k

Question: Does there exist a subset X of S of size k such that all sets of X are pairwise disjoint?



d-Set Packing

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Set Packing admits a kernel with at most $\binom{kd}{d}$ sets and at most $\binom{kd}{d} \cdot d$ elements

d-Set Packing

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Set Packing admits a kernel with at most $\binom{kd}{d}$ sets and at most $\binom{kd}{d} \cdot d$ elements

d-Hitting Set

d-Set Packing

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Set Packing admits a kernel with at most $\binom{kd}{d}$ sets and at most $\binom{kd}{d} \cdot d$ elements

d-Hitting Set

Delete sets, do not turn a no-instance into a yes-instance!

d-Set Packing

Input: A universe U, a family S of sets of size d over U, integer k

Theorem: d-Set Packing admits a kernel with at most $\binom{kd}{d}$ sets and at most $\binom{kd}{d} \cdot d$ elements

d-Hitting Set	d-Set Packing
Delete sets, do not turn a no-instance into a yes-instance!	Delete sets, do not turn a yes-instance into a no-instance!

d-Set Packing: Proof

Theorem: d-Set Packing admits a kernel with at most $\binom{kd}{d}$ sets and at most $\binom{kd}{d} \cdot d$ elements

d-Set Packing: Proof

Theorem: d-Set Packing admits a kernel with at most $\binom{kd}{d}$ sets and at most $\binom{kd}{d} \cdot d$ elements

Algorithm: Compute a (k-1)d-representative set S' of S, delete all elements not in S'.

d-Set Packing: Proof

Theorem: d-Set Packing admits a kernel with at most $\binom{kd}{d}$ sets and at most $\binom{kd}{d} \cdot d$ elements

Algorithm: Compute a (k-1)d-representative set S' of S, delete all elements not in S'.

Compute representation matrix of uniform matroid of rank kd

d-Set Packing: Proof

Theorem: d-Set Packing admits a kernel with at most $\binom{kd}{d}$ sets and at most $\binom{kd}{d} \cdot d$ elements

Algorithm: Compute a (k-1)d-representative set S′ of S, delete all elements not in S′.

Compute representation matrix of uniform matroid of rank kd

$|S'| \leq \binom{kd}{d}$ and $|U'| \leq \binom{kd}{d} \cdot d$

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

-> Suppose S has k disjoint sets. Take family F with the maximum number of sets from S'.

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

-> Suppose S has k disjoint sets. Take family F with the maximum number of sets from S'.

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

-> Suppose S has k disjoint sets. Take family F with the maximum number of sets from S'.

If there is A in F that is not in S', take B be the set of all elements from $F \setminus A$

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

-> Suppose S has k disjoint sets. Take family F with the maximum number of sets from S'.

If there is A in F that is not in S', take B be the set of all elements from $F \setminus A$

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

-> Suppose S has k disjoint sets. Take family F with the maximum number of sets from S'.

If there is A in F that is not in S', take B be the set of all elements from $F \setminus A$

Then $|B|=(k-1)d$ and A fits B.

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

-> Suppose S has k disjoint sets. Take family F with the maximum number of sets from S'.

If there is A in F that is not in S', take B be the set of all elements from $F \setminus A$

Then $|B|=(k-1)d$ and A fits B.

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

-> Suppose S has k disjoint sets. Take family F with the maximum number of sets from S'.

If there is A in F that is not in S', take B be the set of all elements from $F \setminus A$

Then $|B|=(k-1)d$ and A fits B.

Then there is A' in S' that also fits B. But $F \setminus A+A'$ are k disjoint sets contradicting the choice of F.

Correctness

Why (U, S, k) and (U', S', k) are equivalent?

<- Every k disjoint sets of S', are also disjoint in S.

-> Suppose S has k disjoint sets. Take family F with the maximum number of sets from S'.

If there is A in F that is not in S', take B be the set of all elements from $F \setminus A$

Then $|B|=(k-1)d$ and A fits B.

Then there is A' in S' that also fits B. But $F \setminus A+A'$ are k disjoint sets contradicting the choice of F.

Further reading

Juhna
Matroids book
PA book
Kernelization (application protrusions, OCT)
add running times

Open problems

- Faster computation of representative sets for linear matroids?
- Faster computation of representative sets for graphic matroids?
- Compute representative sets for uniform matroids in time linear in input + output?
- Compute representative sets for gammoids without matroid representation?