Chapter 1: Getting Started

The Probabilistic Method

Summer 2020

Freie Universität Berlin

Chapter Overview

- Survey quick applications of the basic method to different areas

§1 Unsatisfiable Formulae

Chapter 1: Getting Started
The Probabilistic Method

§2 Prefix-free Codes

Chapter 1: Getting Started
The Probabilistic Method

§3 Sum-free Subsets

Chapter 1: Getting Started
The Probabilistic Method

§4 Schütte Tournaments

Chapter 1: Getting Started
The Probabilistic Method

§5 Ramsey Numbers

Chapter 1: Getting Started
The Probabilistic Method

§1 Unsatisfiable Formulae

Chapter 1: Getting Started

The Probabilistic Method

Boolean Logic

Binary values

- Computers can only talk in 0s and 1s
- In logical applications, we map those to $False$ and $True$

Logical operators

- Can obtain new truth values from old ones

Not: $\neg$ Or: $\vee$ And: $\wedge$

Boolean formulae

- Can build any $True/False$ expression using these operations
- Such a formula is a function $f: \{0,1\}^n \rightarrow \{0,1\}$

Anatomy of a Formula

Every Boolean formula can be written in Conjunctive Normal Form:

Variables

- $x_i \in \{0,1\}$

Literals

- Variable x_i or its negation $\neg x_i$

Clauses

- 'OR' of literals
- e.g.: $x_1 \vee \neg x_2 \vee x_3$

CNF Formula

- 'AND' of several clauses
- e.g.: $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2) \wedge (x_2 \vee x_3 \vee x_4 \vee \neg x_5)$

A Little Complexity

Satisfiability Problem (SAT)

- Given a Boolean formula f, can f ever evaluate to $True$?
- If not, say f is unsatisfiable

Theorem 1.1.1 (Cook, 1971; Levin, 1973)

SAT is NP-Complete, i.e. is probably very difficult.

A universal model

- Most interesting problems can be reduced to SAT instances
- e.g.: Travelling Salesman Problem, Subgraph Isomorphism, Largest Clique

Restricted Formulae

Simplifying the problem

- Perhaps the problem is easier for ‘nice’ formulae
- k-SAT: each clause must have exactly k literals from distinct variables

Theorem 1.1.2 (Karp, 1972)

For all $k \geq 3$, k-SAT is still NP-Complete.

Does size matter?

- Karp $\Rightarrow$ unsatisfiability does not require long clauses
- Does it at least require many clauses? Are short formulae always satisfiable?

Minimum Unsatisfiability

Extremal problem

- How small can an unsatisfiable instance of k-SAT be?

Definition 1.1.3

Given $k \in \mathbb{N}$, let $m_0(k)$ be the minimum $m \in \mathbb{N}$ for which there is an unsatisfiable instance of k-SAT with m clauses.

Small k-SAT is easy

- Can solve instances of k-SAT with $m < m_0(k)$ clauses in constant time!
- (Existential) answer is always: yes (satisfiable)

A First Lower Bound

Lower bounds

- Given any instance of k-SAT with few clauses, need to show it is satisfiable
- First idea: build a satisfying argument greedily

A worked example ($k = 3$)

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_5)$$

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$
?	?	?	?	?

A First Lower Bound

Lower bounds

- Given any instance of k-SAT with few clauses, need to show it is satisfiable
- First idea: build a satisfying argument greedily

A worked example ($k = 3$)

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_5)$$

x_1	x_2	x_3	x_4	x_5
?	?	?	?	?

Step 1

- Select x_1 as the designated variable for the first clause

A First Lower Bound

Lower bounds

- Given any instance of k-SAT with few clauses, need to show it is satisfiable
- First idea: build a satisfying argument greedily

A worked example ($k = 3$)

$$(1 \vee \neg x_2 \vee x_3) \wedge (0 \vee x_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_5)$$

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$
1	$?$	$?$	$?$	$?$

Step 1

- Select x_1 as the designated variable for the first clause
- Set $x_1 = 1$ to satisfy the clause

A First Lower Bound

Lower bounds

- Given any instance of k-SAT with few clauses, need to show it is satisfiable
- First idea: build a satisfying argument greedily

A worked example ($k = 3$)

$$(1 \vee \neg x_2 \vee x_3) \wedge (0 \vee x_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_5)$$

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$
1	?	?	?	?

Step 2

- The second clause is still unsatisfied

A First Lower Bound

Lower bounds

- Given any instance of k-SAT with few clauses, need to show it is satisfiable
- First idea: build a satisfying argument greedily

A worked example ($k = 3$)

$$(1 \vee 0 \vee x_3) \wedge (0 \vee 1 \vee \neg x_4) \wedge (0 \vee \neg x_3 \vee \neg x_5)$$

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$
1	1	?	?	?

Step 2

- The second clause is still unsatisfied
- Select x_2 as its designated variable, and set $x_2 = 1$

A First Lower Bound

Lower bounds

- Given any instance of k-SAT with few clauses, need to show it is satisfiable
- First idea: build a satisfying argument greedily

A worked example ($k = 3$)

$$(1 \vee 0 \vee 0) \wedge (0 \vee 1 \vee \neg x_4) \wedge (0 \vee 1 \vee \neg x_5)$$

x_1	x_2	x_3	x_4	x_5
1	1	0	$?$	$?$

Step 3

- The third clause is still unsatisfied, so we set $x_3 = 0$

A First Lower Bound

Lower bounds

- Given any instance of k-SAT with few clauses, need to show it is satisfiable
- First idea: build a satisfying argument greedily

A worked example ($k = 3$)

$$(1 \vee 0 \vee 0) \wedge (0 \vee 1 \vee \neg x_4) \wedge (0 \vee 1 \vee \neg x_5)$$

x_1	x_2	x_3	x_4	x_5
1	1	0	?	?

Step 3

- The third clause is still unsatisfied, so we set $x_3 = 0$
- This satisfies the formula, so we are done

A First Lower Bound

Proposition 1.1.4

For all $k \in \mathbb{N}$, $m_0(k) > k$.

Proof

- Let f be an arbitrary k-SAT formula with $m \leq k$ clauses
- We use the greedy algorithm, satisfying each clause one at a time
- When dealing with the ith clause, for $1 \leq i \leq m$, either:
  - it is already satisfied by our previous assignments, or
  - we have set at most $i - 1 \leq m - 1 < k$ variables, so there is a free variable to choose
- Hence we can satisfy all the clauses
- Thus f is satisfiable ■

Being Greedy Doesn't Always Pay

What if we have more clauses?

- This greedy algorithm can get stuck

Extending our example

$$(x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor \neg x_4) \land (\neg x_2 \lor \neg x_3 \lor \neg x_5) \land (\neg x_1 \lor \neg x_2 \lor x_3)$$

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$
?	?	?	?	?

Being Greedy Doesn't Always Pay

What if we have more clauses?

- This greedy algorithm can get stuck

Extending our example

$$(1 \lor 0 \lor 0) \land (0 \lor 1 \lor \neg x_4) \land (0 \lor 1 \lor \neg x_5) \land (0 \lor 0 \lor 0)$$

x_1	x_2	x_3	x_4	x_5
1	1	0	?	?

Steps 1-3

- Proceed as before, with the same assignments
- Now the final clause is unsatisfiable

Being Greedy Doesn't Always Pay

What if we have more clauses?

- This greedy algorithm can get stuck

Extending our example

$$(1 \lor 0 \lor 1) \land (0 \lor 1 \lor \neg x_4) \land (0 \lor 0 \lor 1) \land (0 \lor 0 \lor 1)$$

x_1	x_2	x_3	x_4	x_5
1	1	1	?	0

Formula is still satisfiable

- Could have satisfied earlier clauses with different variables

An Unsatisfiable Formula

Intuition

- Clauses with unique variables are can always be satisfied
- Maybe hardest when all clauses share the same variables

Building an unsatisfiable formula

- With k variables, there are 2^k possible inputs and 2^k possible clauses
- Each clause is unsatisfied by a unique input
  - e.g.: $x_1 \vee \neg x_2 \vee \neg x_3$ is not satisfied by $(x_1, x_2, x_3) = (0,1,1)$
- $\Rightarrow$ The formula with all possible clauses is unsatisfiable

Proposition 1.1.5

For all $k \in \mathbb{N}$, $m_0(k) \leq 2^k$.

A Tight Result

Theorem 1.1.6

For all $k \in \mathbb{N}$, $m_0(k) = 2^k$.

Upper bound

- Previous construction

Lower bound

- Need to show every k-SAT instance with $m < 2^k$ clauses is satisfiable

Existential reformulation

- Given: k-SAT formula f with n variables and $m < 2^k$ clauses
- Goal: show there is some $\vec{x} \in \Omega = \{0,1\}^n$ with the property $f(\vec{x}) = 1$

Randomness to the Rescue

Theorem 1.1.6

For all $k \in \mathbb{N}$, $m_0(k) = 2^k$.

Existential reformulation

- Given: k-SAT formula f with n variables and $m < 2^k$ clauses
- Goal: show there is some $\vec{x} \in \Omega = \{0,1\}^n$ with the property $f(\vec{x}) = 1$

The probabilistic method

- Choose $\vec{x} \in \{0,1\}^n$ uniformly at random
- Show $\mathbb{P}(f(\vec{x}) = 1) > 0$

Bounding Probabilities

Setting

- Given: f, a k-SAT formula with n variables and $m < 2^k$ clauses
- Given: uniformly random $\vec{x} \in \{0,1\}^n$
- Goal: show $\mathbb{P}(f(\vec{x}) = 1) > 0$

Bad events

- Equivalently, want to show $\mathbb{P}(f(\vec{x}) = 0) < 1$
- Let E_i be the event that the ith clause is not satisfied by $\vec{x}$
- $\{f(\vec{x} = 0)\} = \bigcup_{i=1}^{m} E_i$
- $\Rightarrow \mathbb{P}(f(\vec{x}) = 0) = \mathbb{P}(\bigcup_{i=1}^{m} E_i)$

Union Bound

Given arbitrary events E_i, we have $\mathbb{P}(\bigcup_i E_i) \leq \sum_i \mathbb{P}(E_i)$.

Completing the Proof

Individual clauses

- Recall: E_i is the event that the ith clause is not satisfied by $\vec{x}$
- E_i only depends on the values of the k variables it contains
- Exactly one of the 2^k possible values does not satisfy the clause
- $\Rightarrow \mathbb{P}(E_i) = 2^{-k}$

$$\mathbb{P}(f(\vec{x}) = 0) = \mathbb{P}\left(\bigcup_{i=1}^{m} E_i\right) \leq \sum_{i=1}^{m} \mathbb{P}(E_i) = m2^{-k} < 1$$

Conclusion

- Therefore $\mathbb{P}(f(\vec{x}) = 1) = 1 - \mathbb{P}(\vec{x} = 0) > 0$
- Hence there is some $\vec{x} \in \{0,1\}^n$ for which $f(\vec{x}) = 1$ $\blacksquare$

Is Repetition Necessary?

Trivial unsatisfiability

- In our construction to show $m_0(k) \leq 2^k$, each clause had the same variables
- Clauses are then forced to be in conflict with one another

Non-repetitive formulae

- A k-SAT formula is non-repetitive if each clause has a distinct set of variables
- e.g.: cannot have both $(x_1 \vee \neg x_2 \vee x_4)$ and $(\neg x_1 \vee \neg x_2 \vee \neg x_4)$ as clauses

Extremal problem

- How many variables must an unsatisfiable non-repetitive k-SAT formula have?

Definition 1.1.7

Given $k \in \mathbb{N}$, let $n_0(k)$ be the minimum $n \in \mathbb{N}$ for which there is an unsatisfiable non-repetitive k-SAT formula with n variables.

A Lower Bound

Observation

- A non-repetitive k-SAT formula with n variables can have at most $\binom{n}{k}$ clauses

Theorem 1.1.6

For all $k \in \mathbb{N}$, $m_0(k) = 2^k$.

Corollary 1.1.8

For all $k, n \in \mathbb{N}$, if $\binom{n}{k} < 2^k$, then $n_0(k) > n$.

An Upper Bound

Existential formulation

- Set of objects Ω: non-repetitive k-SAT formulae with n variables
- Desired property $\mathcal{P}$: $\forall \vec{x} \in \{0,1\}^n, f(\vec{x}) = 0$

Probabilistic approach

- There are $\binom{n}{k}$ sets of k variables:
  - For each variable x_i, there are two possible literals: x_i and $\neg x_i$
  - Total of 2^k possible clauses for this set of variables
  - Choose one uniformly at random
  - Make these choices independently
- This gives us a random $f \in \Omega$
- Want to show $\mathbb{P}(\forall \vec{x} \in \{0,1\}^n, f(\vec{x}) = 0) > 0$

Analysing the Bad Events

Satisfying assignments

- For each $\vec{x} \in \{0,1\}^n$, let $E_{\vec{x}}$ be the event that $f(\vec{x}) = 1$
- We want to show $\mathbb{P}\left(\bigcup_{\vec{x}} E_{\vec{x}}\right) < 1$
- Union bound:
  - There are 2^n possible $\vec{x}$
  - $\mathbb{P}\left(\bigcup_{\vec{x}} E_{\vec{x}}\right) \leq \sum_{\vec{x}} \mathbb{P}(E_{\vec{x}})$
- Suffices to have $\mathbb{P}(E_{\vec{x}}) < 2^{-\mathrm{n}}$ for all $\vec{x}$

Fix an assignment $\vec{x}$

- For $f(\vec{x}) = 1$, $\vec{x}$ must satisfy each of the $\binom{n}{k}$ clauses
- Let F_i be the event that $\vec{x}$ satisfies the ith clause
- Then $E_{\vec{x}} = \bigcap_i F_i$

Computing Probabilities

Recall

- f formed by choosing a random clause for each set of variables
- $E_{\vec{x}}$: event that $f(\vec{x}) = 1$; F_i: event that $\vec{x}$ satisfies the ith clause of f
- Suffices to show $\mathbb{P}(E_{\vec{x}}) = \mathbb{P}(\cap_i F_i) < 2^{-n}$

Independence

- Clauses are chosen independently $\Rightarrow$ events F_i are independent
- $\Rightarrow \mathbb{P}(\cap_i F_i) = \prod_i \mathbb{P}(F_i)$

Satisfying a single clause

- Given i and our fixed $\vec{x}$, unique choice of literals such that F_i doesn't hold
- $\Rightarrow \mathbb{P}(F_i) = 1 - 2^{-k}$

Putting it all together

A final calculation

- We therefore have $\mathbb{P}(E_{\vec{x}}) = \prod_i \mathbb{P}(F_i) = \left(1 - 2^{-k}\right)^{\binom{n}{k}}$

Exponential bound

For all $x \in \mathbb{R}$, $1 + x \leq e^x$

- $\Rightarrow \mathbb{P}(E_{\vec{x}}) = \left(1 - 2^{-k}\right)^{\binom{n}{k}} \leq e^{-2^{-k}\binom{n}{k}}$
- This is less than 2^{-n} if $\binom{n}{k} > 2^k n \ln 2$

Theorem 1.1.9

For all $k \in \mathbb{N}$, if $\binom{n}{k} < 2^k$, then $n_0(k) > n$, and if $\binom{n}{k} > 2^k n \ln 2$, then $n_0(k) \leq n$.

Just Kidding, There's One More Calculation

Theorem 1.1.9

For all $k \in \mathbb{N}$, if $\binom{n}{k} < 2^k$, then $n_0(k) > n$, and if $\binom{n}{k} > 2^k n \ln 2$, then $n_0(k) \leq n$.

Binomial estimates

- For all $1 \leq k \leq n$, we have $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$
- If $k = \alpha n$, then $\binom{n}{k} = 2^{(1+o(1))H(\alpha)n}$ as $n \to \infty$
  - Binary entropy: $H(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$
- For more estimates:
  http://page.mi.fu-berlin.de/shagnik/notes/binomials.pdf

Just Kidding, There's One More Calculation

Lower bound

- We use $\binom{n}{\alpha n} = 2^{(1+o(1))H(\alpha)n}$
- Therefore $\binom{n}{k} \sim 2^k$ if $k = \alpha n$ and $H(\alpha) = \alpha$
  - This happens for $\alpha = 0.7729 \ldots$

Upper bound

- Binomial coefficient grows very fast
  - $\binom{n+1}{k} = \frac{n+1}{n-k+1}\binom{n}{k} \sim \frac{1}{1-\alpha}\binom{n}{k}$
- $\Rightarrow$ for some constant c, if $n' = \alpha^{-1}k + c\log k$, then $\binom{n'}{k} \sim 2^k n \ln 2$

Corollary 1.1.10

As $k \to \infty$, $n_0(k) = \big(1.2938 \ldots + o(1)\big)k.$

Any questions?

§2 Prefix-free Codes

Chapter 1: Getting Started

The Probabilistic Method

A Motivating Example

Evolutionary pitfalls

- Imagine in some parallel universe a species evolves so that:
  - they develop binary computers, and
  - they communicate their emotional states through a set of five emojis

Technical problem

- How should a binary computer transmit these states?

A First Attempt

Binary encoding

- We can index the emojis with integers $0 - 4$
- The integers $0 - 7$ can be written as binary strings of length 3
- Computers can send these strings to represent the emojis

000 001 010 011 100

Examples

 → 001 | 000 | 100 → 001000100

011000 → 011 | 000 → 

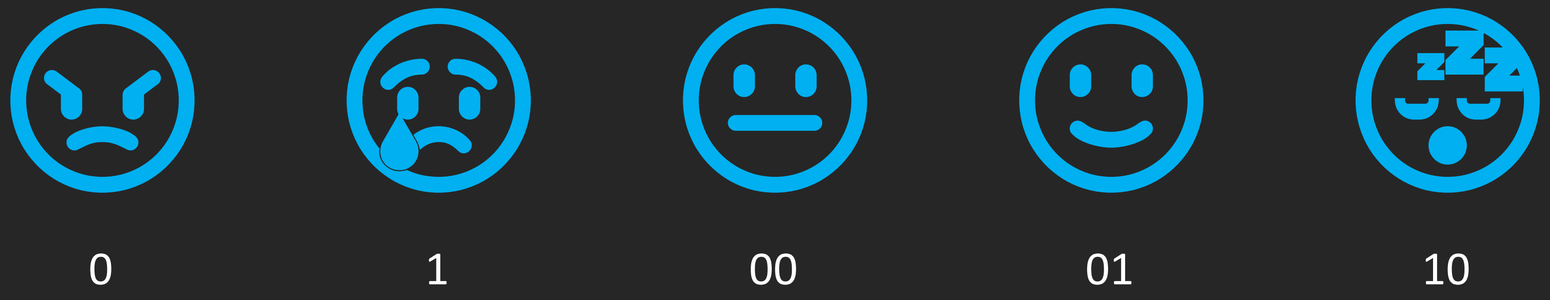
A Problem and a Fix

Wasteful encoding

- This is a bit costly – it takes three bits per emoji
- Can we reduce the bandwidth by using a shorter encoding?

Idea

- We need to encode five emojis
- There are six non-empty binary strings of length at most two
- Five is less than six



The Problem in the “Fix”

😠	😢	😐	🙂	😴
0	1	00	01	10

Encoding is simple

🙂🙂🙂 → 01 | 01 | 01 → 010101

But how do we decode?

010101 → 01 | 01 | 01 → 🙂🙂🙂 ?

010101 → 0 | 1 | 0 | 1 | 0 | 1 → 😠😢😠😢😠😢?

010101 → 01 | 0 | 10 | 1 → 🙂😠😴😢?

Coding: General Framework

Set-up

- Have an alphabet $A = \{a_1, a_2, \ldots, a_n\}$ of size n
- Want to encode the letters of the alphabet as binary strings

Encoding

- Represent each a_i with a word $w_i \in \{0,1\}^*$, a non-empty finite binary string
- Let $\ell_i = |w_i|$ be the length of the word w_i

Objectives

- Decipherability: given a concatenation of words, should be able to recover the original words uniquely
- Efficiency: would like to make the lengths ℓ_i as small as possible

Prefix-free Codes

Prefixes

- Given a word $w \in \{0,1\}^*$, the ℓ-prefix of w is the subword of the first ℓ bits
  - e.g. the non-empty prefixes of $w = 00101$ are 0, 00, 001, 0010 and 00101
  - but the substring 010 is not a prefix

Prefix-free codes

- We say a code from an alphabet A to $\{0,1\}^*$ is prefix-free if no codeword w_i is a prefix of any other codeword w_j, $j \neq i$
- Equivalently, if we place the codewords in the (infinite) binary tree of $\{0,1\}^*$, no codeword is an ancestor of another

Decipherability

Proposition 1.2.1

All prefix-free codes are decipherable.

Proof

- Want to show that the concatenation $w = w_{i_1} w_{i_2} \dots w_{i_s}$ can be decoded
- Base case: $s = 0$
  - In this case, w is the empty string ⇒ no codewords
- Induction step: $s \geq 1$
  - Start from the beginning of w, and read until the prefix is some codeword w_j
  - Must terminate, as w_{i_1} is a prefix of w
  - Cannot terminate on another codeword, as otherwise w_j would be a prefix of w_{i_1}
  - Thus we know a_{i_1} is the first letter
  - Remove w_{i_1}, and decode $w' = w_{i_2} w_{i_3} w_{i_4} \dots w_{i_s}$ (induction hypothesis) ■

Examples

Uniform codes

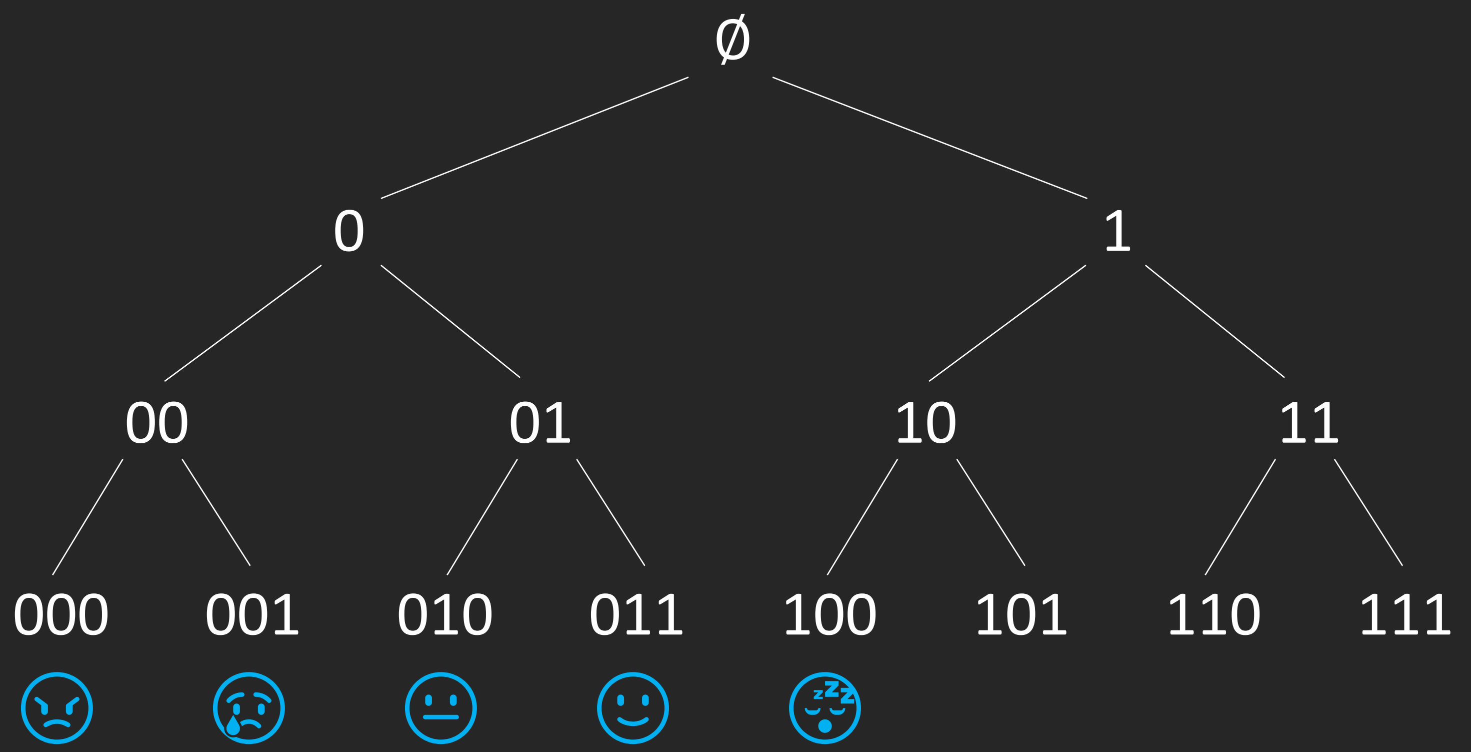
Given ℓ, any injection $A \to \{0,1\}^\ell$ is prefix-free

Length

We must have $|A| \le \left|\{0,1\}^\ell\right| = 2^\ell$

$\Rightarrow \ell \ge \log|A|$

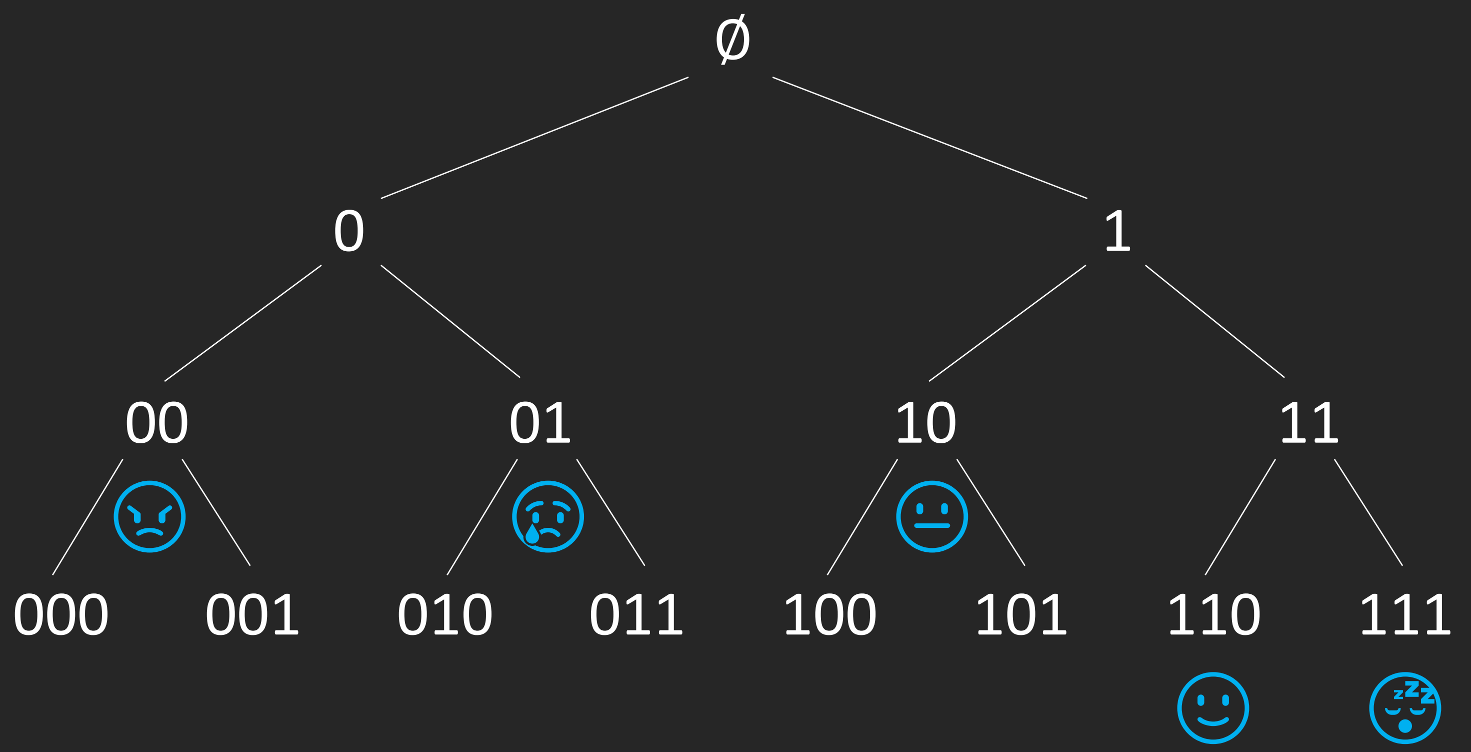
$\Rightarrow \ell \ge \lceil \log|A| \rceil$



Examples - II

Improvements

Can sometimes find prefix-free codes with a shorter average codeword length



Short Prefix-free Codes

Extremal problem

- How small can the average length of a codeword of a prefix-free code be?

Theorem 1.2.2 (Kraft, 1949)

Given an alphabet A of size n, any prefix-free code with codeword lengths $\ell_1, \ell_2, \dots, \ell_n$ must satisfy

$$\sum_{i=1}^{n} 2^{-\ell_i} \leq 1.$$

Corollary 1.2.3 (Convexity)

Given an alphabet A of size n, the average length of the codewords in any prefix-free code is at least $\log n$.

Proof Idea

Existential reformulation

- Want to show that an encoding with shorter codewords is not prefix-free
- Given:
  - an encoding $w_1, w_2, \dots, w_n$ of A with lengths $\ell_1, \ell_2, \dots, \ell_n$ such that $\sum_{i=1}^{n} 2^{-\ell_i} > 1$
- Seek:
  - Codewords w_i, w_j, $i \neq j$, such that w_i is a prefix of w_j

Key observation

- Suppose we have a string $w \in \{0,1\}^*$ such that both w_i, w_j are prefixes of w
  - Then w_i is a prefix of w_j or $\mathrm{w_j}$ is a prefix of w_i

New objective

- Find a string $w \in \{0,1\}^*$ that contains at least two codewords as prefixes

Probabilistic Framework

Probability space

- Let $L = \max\{l_i : i \in [n]\}$ be the length of the longest codeword
- Let $w \in \{0,1\}^L$ be a uniformly random string of length L

Random variables

- Let $X = |\{i : w_i \text{ is a prefix of } w\}|$ count the number of codeword prefixes

Basic Fact

For any random variable X, the events $\{X \geq \mathbb{E}[X]\}$ and $\{X \leq \mathbb{E}[X]\}$ must occur with positive probability.

Simpler objective

- Since X is integer-valued, it suffices to show $\mathbb{E}[X] > 1$

Computing the Expectation

Indicator random variables

- For each $i \in [n]$, let E_i be the event that w_i is a prefix of the random string w
- Let $X_i = 1_{E_i}$ be the indicator function of this event
- Then $X = \sum_{i=1}^{n} X_i$

Linearity of Expectation

For any sequence $X_1, X_2, \dots, X_n$ of random variables, and any sequence $c_1, c_2, \dots, c_n$ of constants, if $X = c_1 X_1 + c_2 X_2 + \cdots + c_n X_n$, then

$$\mathbb{E}[X] = c_1 \mathbb{E}[X_1] + c_2 \mathbb{E}[X_2] + \cdots + c_n \mathbb{E}[X_n].$$

Reduction to probabilities

- We therefore have $\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} \mathbb{P}(E_i)$

Finishing the Proof

Recall

- w is a uniformly random string
- X is the number of codewords that are prefixes of w
- E_i is the event that the codeword w_i is a prefix of w
- $\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{P}(E_i)$

Computing probabilities

- The event E_i only depends on the first ℓ_i bits of w
- This is a uniformly random string in $\{0,1\}^{\ell_i}$
- $\Rightarrow \mathbb{P}(E_i) = 2^{-\ell_i}$

A grand finale

- $\Rightarrow$ if $\mathbb{E}[X] = \sum_{i=1}^{n} 2^{-\ell_i} > 1$, there is some $w \in \{0,1\}^*$ with two codewords as prefixes
- Hence, in any prefix-free code, $\sum_{i=1}^{n} 2^{-\ell_i} \leq 1$ $\blacksquare$

Linearity of Expectation

Union bound revisited

- In the previous calculation, we saw the expression $\sum_i \mathbb{P}(E_i)$
- Union bound: $\mathbb{P}(\cup_i E_i) < \sum_i \mathbb{P}(E_i)$
- $\therefore \sum_i \mathbb{P}(E_i) < 1 \Rightarrow$ with positive probability, none of the events E_i occur

Using linearity instead

- $\sum_i \mathbb{P}(E_i)$ is the expectation of the number X of events E_i that occur
- $\therefore \sum_i \mathbb{P}(E_i) < 1 \Rightarrow$ with positive probability, $X = 0$
- With linearity, we get information when $\sum_i \mathbb{P}(E_i) \geq 1$ as well

Any questions?

§3 Sum-free Subsets

Chapter 1: Getting Started

The Probabilistic Method

Sum Theorems

Definition 1.3.1

A set A is sum-free if there are no $x, y, z \in A$ with $x + y = z$.

Theorem 1.3.2 (Fermat, 1637; Wiles, 1995)

For all $n \geq 3$, the set $\{x^n : x \in \mathbb{N}\}$ is sum-free.

Theorem 1.3.3 (Schur, 1912)

$\mathbb{N}$ cannot be partitioned into finitely many sum-free sets.

Sum-free Subsets of $[n]$

Question

How large can a sum-free subset of $[n]$ be?

Answer

- If A is sum-free, then $|\mathrm{A}| \leq \left\lceil \frac{n}{2} \right\rceil$
- Odd integers: $O = \{x \in [n]: x \equiv 1 \pmod 2\}$
- Large integers: $L = \left\{x \in [n]: x > \frac{n}{2}\right\}$
- These are the only two maximum sum-free subsets of $[n]$

Sum-free Subsets of Sets

Theorem 1.3.4 (Deshouillers, Freiman, Sós)

If $A \subseteq [n]$ is sum-free, then either $A \subseteq O$, $A \subseteq L$, or $|A| < \frac{2}{5}n + 1$.

Question (Erdős, 1965)

Does every set of n natural numbers have a large sum-free subset?

Extremal function

- Given a set $S \subseteq \mathbb{N}$, let $f(S) = \max\ \{|A|: A \subseteq S, A \text{ sum−free}\}$
- Let $f(n) = \min\ \{f(S): S \subset \mathbb{N}, |S| = n\}$
- Question: how quickly does $f(n)$ grow?

Upper Bounds

A trivial bound

- $f(n) \leq f([n]) = \left\lceil \frac{n}{2} \right\rceil$
- Any good set should have lots of (well-distributed) sums
- $[n]$ has lots of sums – could this be best possible?

Beating trivial

- Recall: biggest sum-free subsets have odd or large integers
- Let $T \subseteq [n]$ be a set of $\frac{n}{10}$ large odd integers, take $S = [n] \setminus T$
- If $A \subseteq S$ is sum-free, then either $A \subseteq O \setminus T$, $A \subseteq L \setminus T$ or $|A| < \frac{2}{5}n + 1$
- Thus $f\left(\frac{9}{10}n\right) \leq f(S) < \frac{2}{5}n + 1$
- $\Rightarrow f(n) \leq \frac{4}{9}n + \frac{10}{9}$

Lower Bounds

Goal

- Given a set S of n natural numbers, find a large sum-free $A \subseteq S$

Greedy approach

- Start with $A = \emptyset$, and add elements one-by-one, keeping A sum-free
- If $|A| = a$, A defines at most $\binom{a+1}{2}$ sums
- If $\binom{a+1}{2} < n - a$, there is an element of $S \setminus A$ that can be added to A
- $\Rightarrow f(n) > \sqrt{2n} - 2$

Theorem 1.3.5 (Erdős, 1965)

For all $n \in \mathbb{N}$, $f(n) \geq \frac{1}{3}(n+1)$.

A Cyclic Digression

The problem with $[n]$

- $[n]$ does have large sum-free sets, O and L
- But S might be far away from these

The cyclic group has more symmetry

- Largest sum-free set in $\mathbb{Z}_p$, p prime?
  - $M = \left\{x : \frac{1}{3}p < x < \frac{2}{3}p\right\}$ is sum-free
  - Cauchy-Davenport: if $A \subseteq \mathbb{Z}_p$, then $|A + A| \geq \min\{2|A| - 1, p\}$
  - Since $A \cap (A + A) = \emptyset$, $|A| \leq \left\lceil \frac{p}{3} \right\rceil$
- $\mathbb{Z}_p$ has many large sum-free sets
  - For any $\alpha \in \mathbb{Z}_p \setminus \{0\}$, $\alpha M = \{\alpha x : x \in M\}$ is also sum-free

Finding Large Sum-free Subsets

Theorem 1.3.5 (Erdős, 1965)

For all $n \in \mathbb{N}$, $f(n) \geq \frac{1}{3}(n+1)$.

Proof idea

- Given a set $S \subset \mathbb{N}$ of size n, embed S in $\mathbb{Z}_p$ for some suitable p
- $\mathbb{Z}_p$ has many large sum-free subsets
  - Find one that intersects S significantly

Randomness to the rescue

- A random sum-free subset works!

Setting Up

Choosing a prime

- Let $p = 3k + 2$ be prime with $p > \max S$
- Then $M = \{k+1, k+2, \dots, 2k+1\}$ is sum-free with size $k + 1$
- Embed $S \subseteq \mathbb{Z}_p$

Choosing a sum-free subset

- Let $\alpha \in \mathbb{Z}_p \setminus \{0\}$ be chosen uniformly at random
- Let $S_\alpha = S \cap \alpha M$
- $S_\alpha \subseteq S$ is sum-free:
  - If $x + y = z$ in S_α, then $x + y \equiv z \pmod{p}$, so this would be a sum in αM

No Devil in the Details

Using linearity

- $|S_\alpha| = \sum_{s\in S} 1_{\{s\in\alpha M\}}$
- $\Rightarrow \mathbb{E}[|S_\alpha|] = \mathbb{E}\left[\sum_{s\in S} 1_{\{s\in\alpha M\}}\right] = \sum_{s\in S} \mathbb{E}\left[1_{\{s\in\alpha M\}}\right] = \sum_{s\in S} \mathbb{P}(s\in\alpha M)$

Computing probabilities

- $s \in \alpha M \Leftrightarrow \alpha^{-1}s \in M$
- α uniform over $\mathbb{Z}_p \setminus \{0\} \Rightarrow \alpha^{-1}$ uniform $\Rightarrow \alpha^{-1}s$ uniform
- $\Rightarrow \mathbb{P}(\alpha^{-1}s \in M) = \frac{|M|}{p-1} = \frac{k+1}{3k+1} > \frac{1}{3}$

Finishing the proof

- $\Rightarrow \mathbb{E}[|S_\alpha|] > \frac{1}{3}|S| \Rightarrow$ for some α, $|S_\alpha| \geq \frac{1}{3}(n+1)$
- This gives a sum-free subset of S of the desired size ■

Finishing the Story

Improving the lower bound

- Using Fourier analysis, Bourgain (1997) proved $f(n) \geq \frac{1}{3}(n+2)$ for $n \geq 3$
- Best-known bound to date

Upper bounds

- Blow-ups of small constructions: several improvements over the years
- Until Eberhard, Green and Manners (2014) proved $f(n) \leq \left(\frac{1}{3} + o(1)\right) n$
  - Construction randomised, but intricate

Any questions?

§4 Schütte Tournaments

Chapter 1: Getting Started

The Probabilistic Method

War by Proxy

Rival superpowers

- Two powerful nations go to war
- Hire private military companies to do the actual fighting

Objectives

- Have enough power
  - Need to ensure that hired companies can defeat any of the companies the enemy hires
- Be economical
  - Hire as few companies as possible

Problem

- How many companies must be hired?

A Graph Theoretic Representation

Tournaments

- Build a directed graph
  - Vertices: private military companies
  - Arcs: edge $x \to y$ if x would defeat y in battle
  - For every pair $\{x, y\}$, exactly one of the arcs $x \to y$ or $y \to x$ is in the graph
- Such a graph is called a tournament

Objectives

- Dominating set
  - A subset of vertices S such that, for every $x \in V \setminus S$, there is some $s \in S$ with $s \to x$
  - Then we can always defeat the enemy's army, regardless of their choice
- Economical
  - Want to choose a small dominating set

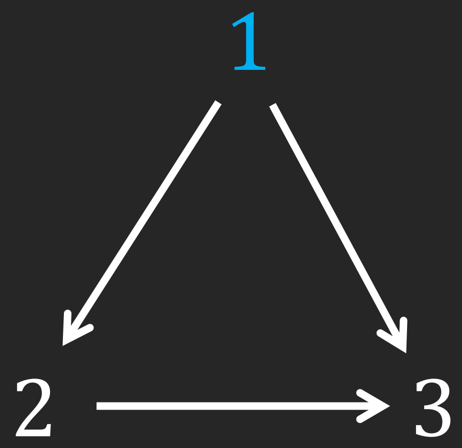
Small Examples

Easy case

$$\begin{tikzcd} & 1 \arrow[dl] \arrow[dr] & \\ 2 \arrow[rr] & & 3 \end{tikzcd}$$

Small Examples

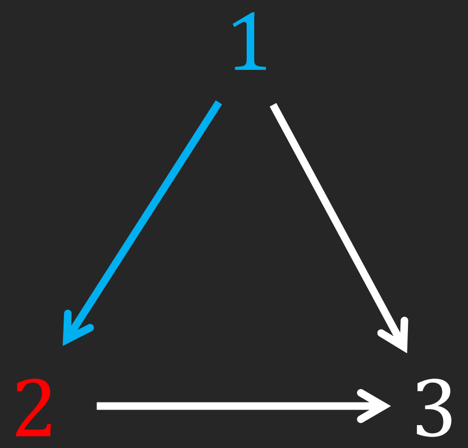
Easy case



- One vertex beats all others

Small Examples

Easy case



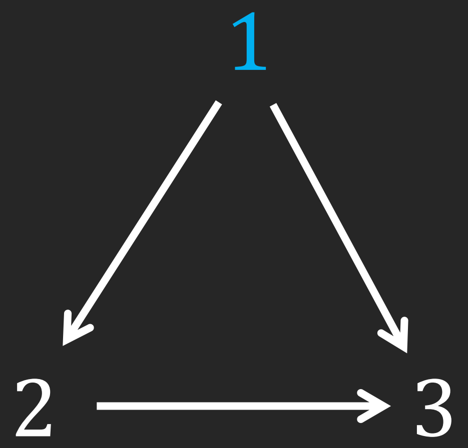
- One vertex beats all others
- Defeats any choice the enemy makes

Small Examples

- One vertex beats all others
- Defeats any choice the enemy makes

Small Examples

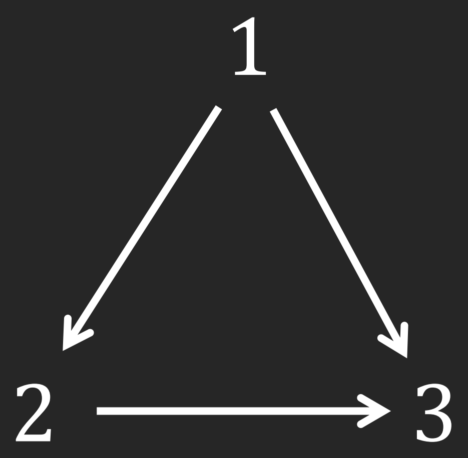
Easy case



- One vertex beats all others
- Defeats any choice the enemy makes
- Dominating set of size one

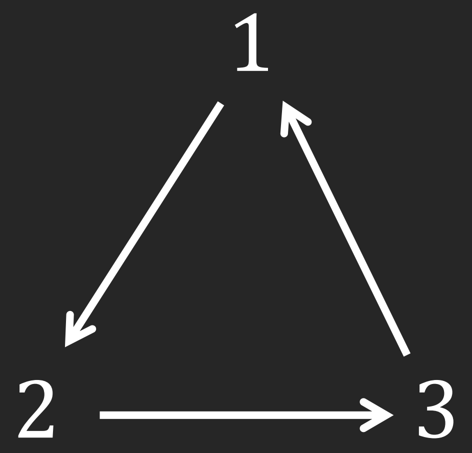
Small Examples

Easy case



- One vertex beats all others
- Defeats any choice the enemy makes
- Dominating set of size one

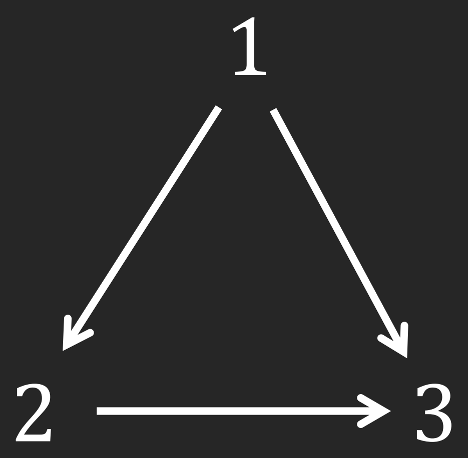
Harder case



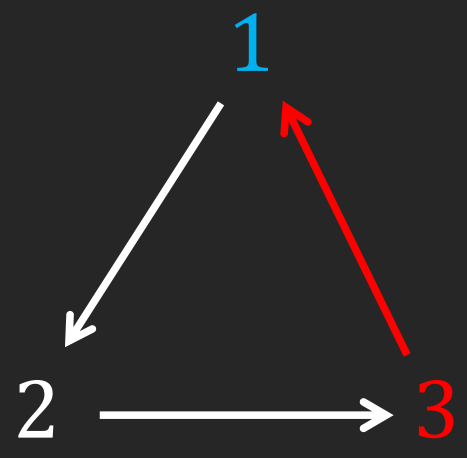
- No such vertex

Small Examples

Easy case



Harder case

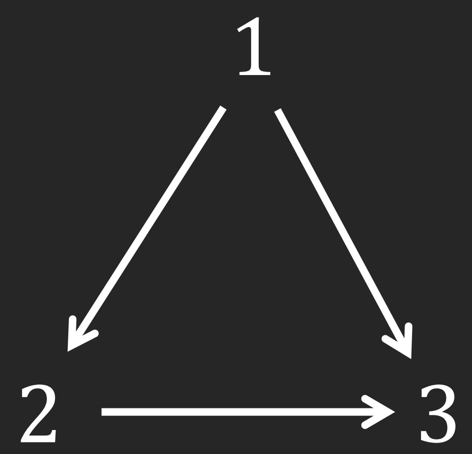


- One vertex beats all others
- Defeats any choice the enemy makes
- Dominating set of size one

- No such vertex
- Any vertex we choose loses to another

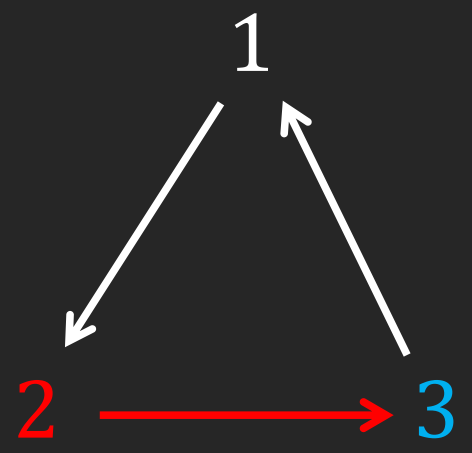
Small Examples

Easy case



- One vertex beats all others
- Defeats any choice the enemy makes
- Dominating set of size one

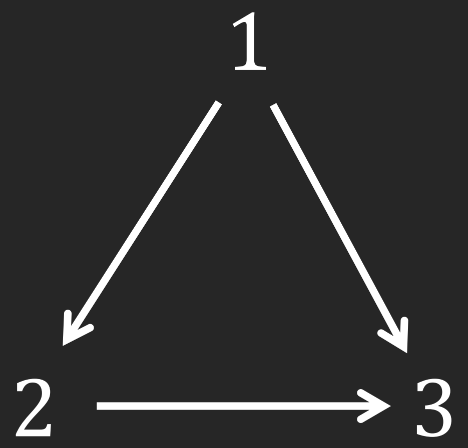
Harder case



- No such vertex
- Any vertex we choose loses to another

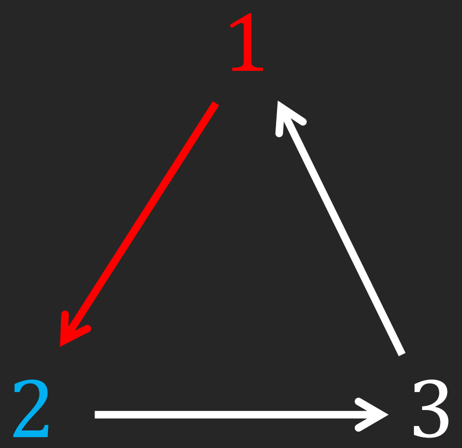
Small Examples

Easy case



- One vertex beats all others
- Defeats any choice the enemy makes
- Dominating set of size one

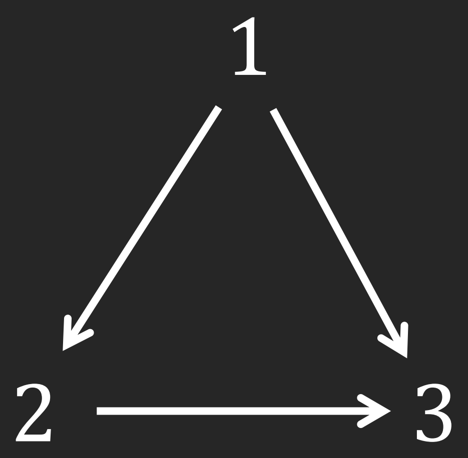
Harder case



- No such vertex
- Any vertex we choose loses to another

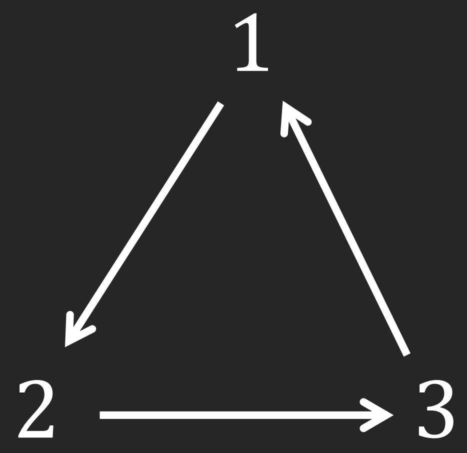
Small Examples

Easy case



- One vertex beats all others
- Defeats any choice the enemy makes
- Dominating set of size one

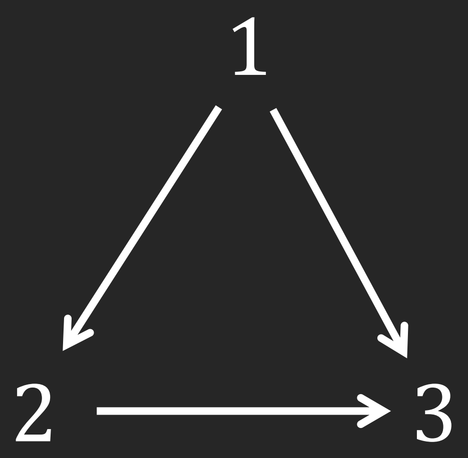
Harder case



- No such vertex
- Any vertex we choose loses to another

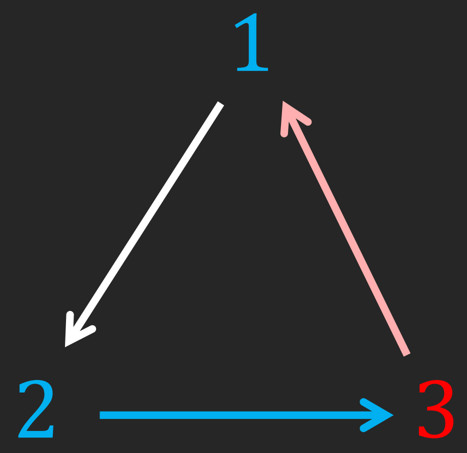
Small Examples

Easy case



- One vertex beats all others
- Defeats any choice the enemy makes
- Dominating set of size one

Harder case



- No such vertex
- Any vertex we choose loses to another
- Dominating set of size two exists

An Extremal Reformulation

Worst-case scenario

- How large can the smallest dominating set in an n-vertex tournament be?
- Inverse formulation
  - Say T has the Schütte property S_k if it has no dominating set of size at most k
  - Let $\sigma(k)$ be the minimum number of vertices in a tournament with the property S_k
  - $\Rightarrow$ if $n < \sigma(k)$, then T has a dominating set of size $\leq k$

Proving bounds on $\sigma(k)$

- Lower bound: $\sigma(k) > n$
  - Prove that any tournament on n vertices has a dominating set of size $\leq k$
- Upper bound: $\sigma(k) \leq n$
  - Prove there is a tournament on n vertices without a dominating set of size $\leq k$

The Greedy Lower Bound

Proposition 1.4.1

For all $k \in \mathbb{N}$, $\sigma(k) \geq 2^{k+1} - 1$.

A recursive algorithm

- Given an optimal tournament T, let $v \in V(T)$
- Let A be the vertices dominating v, and B the vertices v dominates
  - Thus $V(T) = A \cup B \cup \{v\}$, with $A \rightarrow v \rightarrow B$
- Let S' be a dominating set in $T[A]$, and set $S = S' \cup \{v\}$
- If $x \in V(T) \setminus S$:
  - If $x \in A$, then x is dominated by S', so there is an $s \in S' \subseteq S$ with $s \rightarrow x$
  - If $x \notin A$, then $x \in B$, so $v \rightarrow x$
- Thus S is a dominating set for T

Choosing the Right Vertex

Large out-degree

- If A is small, then it has a small dominating set
- Thus we should choose v to make A as small as possible
- $\Rightarrow$ choose a vertex of maximum out-degree
  - Average out-degree is $\frac{1}{n}\binom{n}{2} = \frac{1}{2}(n-1)$
  - $\Rightarrow$ by choosing v of maximum out-degree, we ensure $|A| \leq \frac{1}{2}(n-1)$

Induction

- Since T has the property S_k, $T[A]$ must have the property S_{k-1}
- $\Rightarrow \frac{1}{2}(n-1) \geq |A| \geq \sigma(k-1) \geq 2^k - 1$ (induction hypothesis)
- Solving gives $\sigma(k) = n \geq 2^{k+1} - 1$ ■

An Indomitable Tournament

Theorem 1.4.2 (Erdős, 1963)

If $\binom{n}{k}\left(1-2^{-k}\right)^{n-k} < 1$, then there is an n-vertex tournament with the property S_k.

Goal

- Need to construct a tournament with no dominating set of size k
- Greedy argument: tournament should be close to regular
- Idea: try a random tournament T

Random tournament

- Vertex set: $V = [n]$
- For every pair $x, y \in [n]$, choose $x \to y$ or $y \to x$ uniformly at random

Disproving Domination

Bad events

- Given a set $S \in \binom{[n]}{k}$, let E_S be the event that S is a dominating set
- Then $\mathbb{P}(T \text{ has property } S_k) = 1 - \mathbb{P}(\bigcup_S E_S) \geq 1 - \sum_S \mathbb{P}(E_S)$
  - Suffices to show $\sum_S \mathbb{P}(E_S) < 1$

Computing probabilities

- Fix $S \in \binom{[n]}{k}$
- For S to dominate a fixed vertex v, cannot have all edges $v \to S$
  - k edges, chosen independently $\Rightarrow$ probability is $1 - 2^{-k}$
- This must be true for all vertices in $V \setminus S$
  - Edges again independent $\Rightarrow \mathbb{P}(E_S) = \left(1 - 2^{-k}\right)^{n-k}$
- $\Rightarrow \sum_S \mathbb{P}(E_S) = \binom{n}{k}\left(1 - 2^{-k}\right)^{n-k} < 1.$ ∎

Computing the Bound

Find the smallest n for which $\binom{n}{k}\left(1-2^{-k}\right)^{n-k} < 1$

- Estimates:
  - $\binom{n}{k} \le n^k$ and $1 - 2^{-k} < e^{-2^{-k}}$
- $\Rightarrow$ suffices to have $n^k e^{-2^{-k}(n-k)} < 1$
  - $\Leftrightarrow k \ln n < (n-k)2^{-k}$ $(*)$
- $(*) \Rightarrow n > 2^k$
  - $\Rightarrow \ln n > k \ln 2$
- $(*) \Rightarrow n > k^2 2^k \ln 2$
  - $\Rightarrow \ln n > k\big(\ln 2 + o(1)\big)$, so this suffices

Corollary 1.4.3

As $k \to \infty$, $\sigma(k) \le k^2 2^k \big(\ln 2 + o(1)\big)$.

Any questions?

§5 Ramsey Numbers

Chapter 1: Getting Started

The Probabilistic Method

Reviewing the Classics

Definition 1.5.1 (Ramsey number)

Given $k \in \mathbb{N}$, $R(k)$ is the minimum n for which any n-vertex graph has either a clique or independent set on k vertices.

Theorem 1.5.2 (Erdős, 1947)

As $k \to \infty$, we have

$$R(k) \geq \left(\frac{1}{e\sqrt{2}} + o(1)\right) k\sqrt{2}^k.$$

Proof idea

- Show that a uniformly random graph on this many vertices works

Ramsey Upper Bounds

Theorem 1.5.3 (Erdős-Szekeres, 1935)

For all $k \in \mathbb{N}$, we have $R(k) \le \binom{2k-2}{k-1}$. In particular, as $k \to \infty$, $R(k) \le \frac{1+o(1)}{4\sqrt{\pi k}} 4^k$.

Proof by induction

- Introduce the asymmetric Ramsey numbers

Definition 1.5.4 (Asymmetric Ramsey numbers)

Given $\ell, k \in \mathbb{N}$, $R(\ell, k)$ is the minimum n for which any n-vertex graph contains either a clique on ℓ vertices or an independent set on k vertices.

Asymmetric Ramsey Bounds

Problem

- For fixed $\ell \in \mathbb{N}$, how does $R(\ell, k)$ grow as $k \to \infty$?

Theorem 1.5.5 (Erdős-Szekeres, 1935)

For all $\ell, k \in \mathbb{N}$,

$$R(\ell, k) \leq \binom{\ell + k - 2}{\ell - 1} = O\left(k^{\ell - 1}\right).$$

- Asymmetric Ramsey numbers grow at most polynomially
- Can we find matching lower bounds?

Turán's Lower Bound

Goal

- Find a K_ℓ-free graph with no large independent sets

Intuition

- More edges $\Rightarrow$ fewer independent sets
- How dense can a K_ℓ-free graph be?

Theorem 1.5.6 (Turán, 1941)

An n-vertex K_ℓ-free graph can have at most $\left(1 - \frac{1}{\ell-1}\right)\binom{n}{2}$ edges.

Construction

- Complete $(\ell - 1)$-partite graph $T_{n,\ell-1}$
- $\alpha\left(T_{n,\ell-1}\right) = \frac{n}{\ell-1} \Rightarrow R(\ell, k) > (\ell - 1)(k - 1)$

What About Randomness?

$R(k)$ lower bound

- Symmetric situation – can switch edges and non-edges
- Used a uniformly random graph
- Equivalently: each edge appears independently with probability $\frac{1}{2}$

$R(\ell, k)$ for fixed $\ell \in \mathbb{N}, k \to \infty$

- Situation far from symmetric
  - “easier” to make clique on ℓ vertices than an independent set on k vertices
- Should focus on graphs with fewer edges

Erdős-Rényi model

- $G(n, p)$: n vertices, each edge appears independently with probability p
- Allows us to “see” sparser graphs

A Random Lower Bound

Theorem 1.5.7

Given $\ell, k, n \in \mathbb{N}$ and $p \in [0,1]$, if

$$\binom{n}{\ell} p^{\binom{\ell}{2}} + \binom{n}{k} (1-p)^{\binom{k}{2}} < 1,$$

then $R(\ell, k) > n$.

Proof idea

- Sample the random graph $G \sim G(n,p)$
- What could go wrong?
  - Could find a clique on ℓ vertices
  - Could find an independent set on k vertices

Analysing Bad Events

Bad cliques

- Given a set $S \in \binom{[n]}{\ell}$, let E_S be the event that $G[S]$ is a clique
- $\binom{\ell}{2}$ pairs, each an edge independently with probability p
- $\Rightarrow \mathbb{P}(E_S) = p^{\binom{\ell}{2}}$

Bad independent sets

- Given a set $T \in \binom{[n]}{k}$, let F_T be the event that $G[T]$ is an independent set
- $\binom{k}{2}$ pairs, each a non-edge independently with probability $1 - p$
- $\Rightarrow \mathbb{P}(F_T) = (1 - p)^{\binom{k}{2}}$

Completing the Proof

Recall

- $E_S = \{G[S] \text{ is an } \ell\text{−clique}\}, \mathbb{P}(E_S) = p^{\binom{\ell}{2}}$
- $F_T = \{G[T] \text{ is an independent } k\text{−set}\}, \mathbb{P}(F_T) = (1-p)^{\binom{k}{2}}$

Union bound does the job

- $\{G \text{ not Ramsey}\} = (\cup_S E_S) \cup (\cup_T F_T)$
- $\therefore \mathbb{P}(G \text{ not Ramsey}) = \mathbb{P}\big((\cup_S E_S) \cup (\cup_T F_T)\big) \leq \sum_S \mathbb{P}(E_S) + \sum_T \mathbb{P}(F_T)$
- $\sum_S \mathbb{P}(E_S) + \sum_T \mathbb{P}(F_T) = \binom{n}{\ell} p^{\binom{\ell}{2}} + \binom{n}{k}(1-p)^{\binom{k}{2}} < 1$

$\Rightarrow \mathbb{P}(G \text{ Ramsey}) = 1 - \mathbb{P}(G \text{ not Ramsey}) > 0$ ∎

An Actual Bound

Theorem 1.5.7

Given $\ell, k, n \in \mathbb{N}$ and $p \in [0,1]$, if
$$\binom{n}{\ell} p^{\binom{\ell}{2}} + \binom{n}{k} (1-p)^{\binom{k}{2}} < 1,$$
then $R(\ell, k) > n$.

What does this tell us about $R(\ell, k)$?

Goal

- Maximise n
- Subject to $\binom{n}{\ell} p^{\binom{\ell}{2}} + \binom{n}{k} (1-p)^{\binom{k}{2}} < 1$ for some $p \in [0,1]$

Computing a Lower Bound

Goal

- Maximise n
- Subject to $\binom{n}{\ell}p^{\binom{\ell}{2}} + \binom{n}{k}(1-p)^{\binom{k}{2}} < 1$ for some $p \in [0,1]$

Varying p

- As p increases, $\binom{n}{\ell}p^{\binom{\ell}{2}}$ increases and $\binom{n}{k}(1-p)^{\binom{k}{2}}$ decreases
- $\Rightarrow$ at optimum, expect both quantities to be comparable

Simplification

- Instead solve $\binom{n}{\ell}p^{\binom{\ell}{2}} < \frac{1}{2}$ and $\binom{n}{k}(1-p)^{\binom{k}{2}} < \frac{1}{2}$

Computing Some More

$\binom{n}{\ell}p^{\binom{\ell}{2}} < \frac{1}{2}$:

- Bound $\binom{n}{\ell} \leq n^{\ell}$, so $\binom{n}{\ell}p^{\binom{\ell}{2}} \leq \left(np^{\frac{\ell-1}{2}}\right)^{\ell}$
- Sufficient to have $p \leq \left(1 - o(1)\right)n^{-2/(\ell-1)}$

$\binom{n}{k}(1-p)^{\binom{k}{2}} < \frac{1}{2}$:

- Bound $\binom{n}{k} \leq n^{k}$ and $1 - p \leq e^{-p}$, so $\binom{n}{k}(1-p)^{\binom{k}{2}} \leq \left(ne^{-p(k-1)/2}\right)^{k}$
- Suffices to have $ne^{-p(k-1)/2} < 1 \Rightarrow p(k-1) > 2\ln n$
- Substitute $p \approx n^{-2/(\ell-1)}$
- $\Rightarrow k > 2n^{2/(\ell-1)}\ln n$

Concluding the Computations

Recall

- $k \approx 2n^{2/(\ell-1)} \ln n$

Solve for n

- $n \approx \left(\frac{k}{2 \ln n}\right)^{\frac{\ell-1}{2}} \approx \left(\frac{k}{2 \ln k}\right)^{\frac{\ell-1}{2}}$

Corollary 1.5.8

For fixed $\ell \in \mathbb{N}$ and $k \to \infty$, we have

$$\Omega\left(\left(\frac{k}{2 \ln k}\right)^{\frac{\ell-1}{2}}\right) = R(\ell, k) = O\left(k^{\ell-1}\right).$$

Any questions?