Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Lehrstuhl für
INFORMATIK I
Algorithmen & Komplexität

Institut für Informatik

Exact Algorithms

Sommer Term 2020

Lecture 5. Inclusion–Exclusion

Based on: [Exact Exponential Algorithms: §4]

Further reading: [Parameterized Algorithms: §10.1]

(slides by J. Spoerhase, Th. van Dijk, S. Chaplick, and A. Wolff)

Alexander Wolff

Lehrstuhl für Informatik I

Puzzle

How many numbers ≤ 1000 are *not* divisible by 2, 3 or 5?

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, $\dots$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

$P_2 = \{n \in \mathcal{U} \mid n \bmod 2 \equiv 0\}$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

$P_2 = \{n \in \mathcal{U} \mid n \bmod 2 \equiv 0\}$ $\qquad |P_2| = 500$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

$P_2 = \{n \in \mathcal{U} \mid n \text{ mod } 2 \equiv 0\}$ $\qquad |P_2| = 500$

$P_3 = \{n \in \mathcal{U} \mid n \text{ mod } 3 \equiv 0\}$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

$$P_2 = \{n \in \mathcal{U} \mid n \bmod 2 \equiv 0\} \qquad |P_2| = 500$$

$$P_3 = \{n \in \mathcal{U} \mid n \bmod 3 \equiv 0\} \qquad |P_3| = 333$$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

$P_2 = \{n \in \mathcal{U} \mid n \text{ mod } 2 \equiv 0\}$ $\qquad |P_2| = 500$

$P_3 = \{n \in \mathcal{U} \mid n \text{ mod } 3 \equiv 0\}$ $\qquad |P_3| = 333$

$P_5 = \{n \in \mathcal{U} \mid n \text{ mod } 5 \equiv 0\}$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

$P_2 = \{n \in \mathcal{U} \mid n \text{ mod } 2 \equiv 0\}$ $\quad |P_2| = 500$

$P_3 = \{n \in \mathcal{U} \mid n \text{ mod } 3 \equiv 0\}$ $\quad |P_3| = 333$

$P_5 = \{n \in \mathcal{U} \mid n \text{ mod } 5 \equiv 0\}$ $\quad |P_5| = 200$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

$P_2 = \{n \in \mathcal{U} \mid n \text{ mod } 2 \equiv 0\}$ $\qquad |P_2| = 500$

$P_3 = \{n \in \mathcal{U} \mid n \text{ mod } 3 \equiv 0\}$ $\qquad |P_3| = 333$

$P_5 = \{n \in \mathcal{U} \mid n \text{ mod } 5 \equiv 0\}$ $\qquad |P_5| = 200$

Goal: $|\mathcal{U}| - |P_2 \cup P_3 \cup P_5|$

Puzzle

How many numbers ≤ 1000 are not divisible by 2, 3 or 5?

Universe: $\mathcal{U} = [1000] := \{n \in \mathbb{N} \colon 1 \leq n \leq 1000\}$

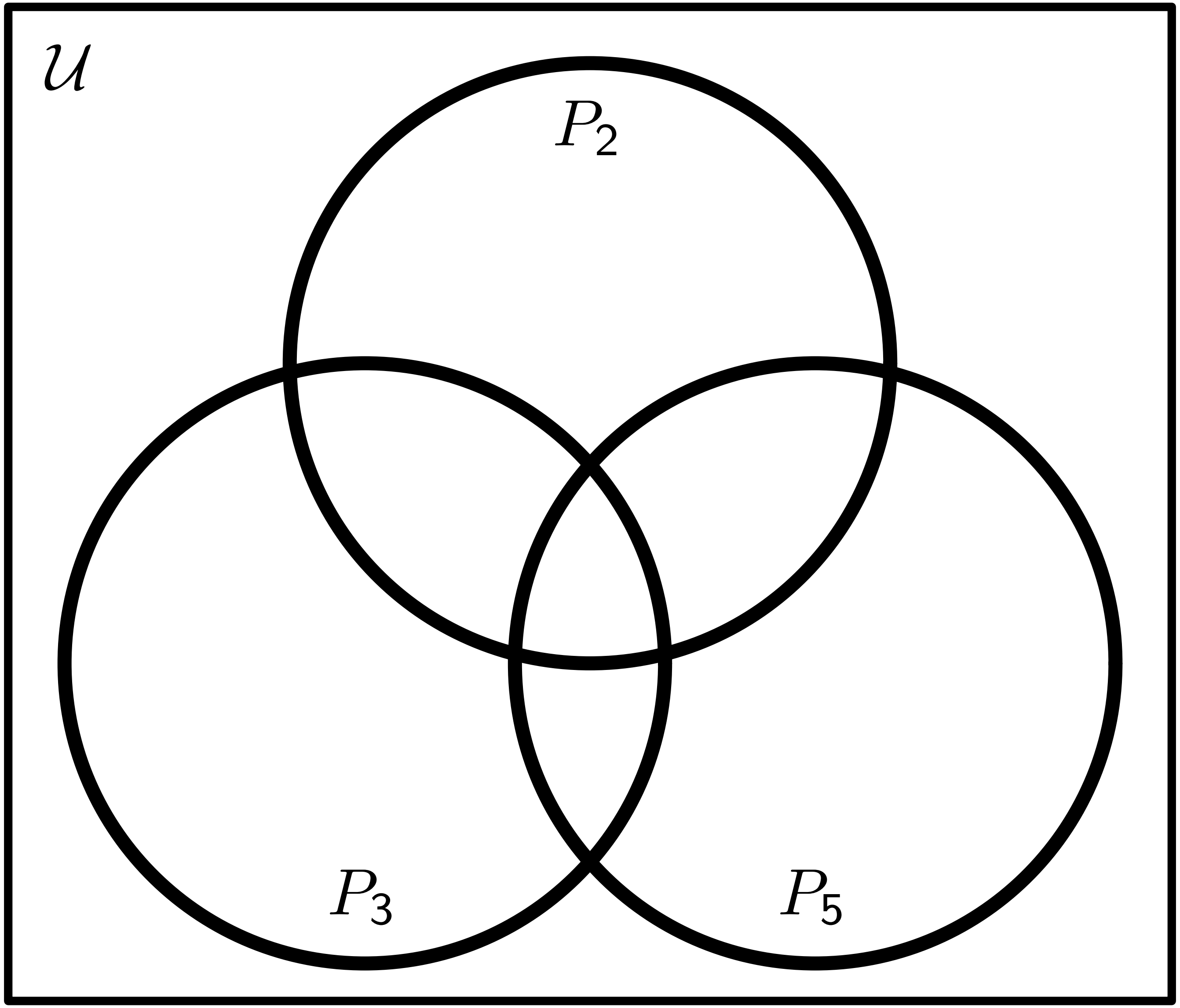
$P_2 = \{n \in \mathcal{U} \mid n \text{ mod } 2 \equiv 0\}$ $|P_2| = 500$

$P_3 = \{n \in \mathcal{U} \mid n \text{ mod } 3 \equiv 0\}$ $|P_3| = 333$

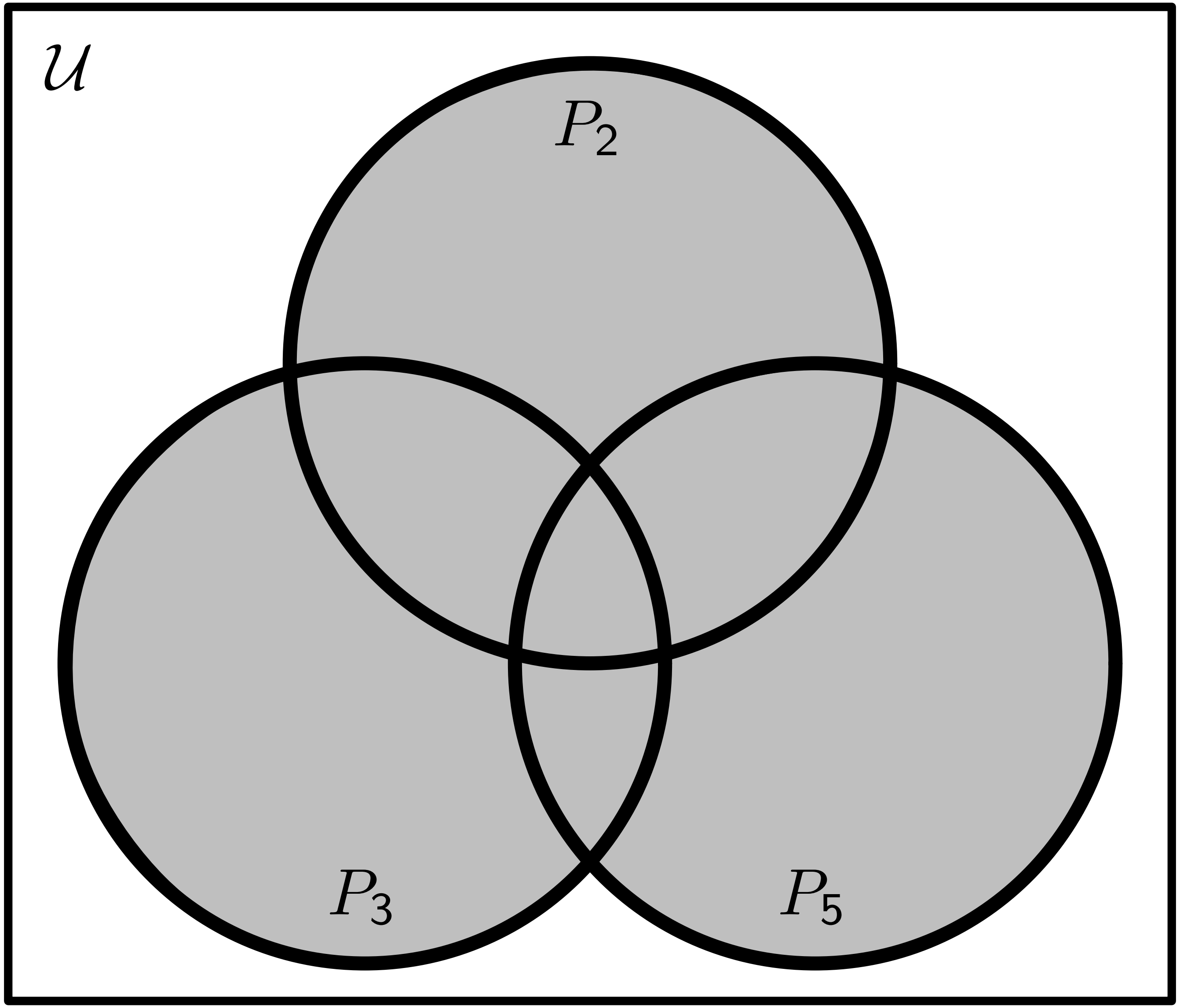
$P_5 = \{n \in \mathcal{U} \mid n \text{ mod } 5 \equiv 0\}$ $|P_5| = 200$

Goal: $|\mathcal{U}| - |P_2 \cup P_3 \cup P_5|$

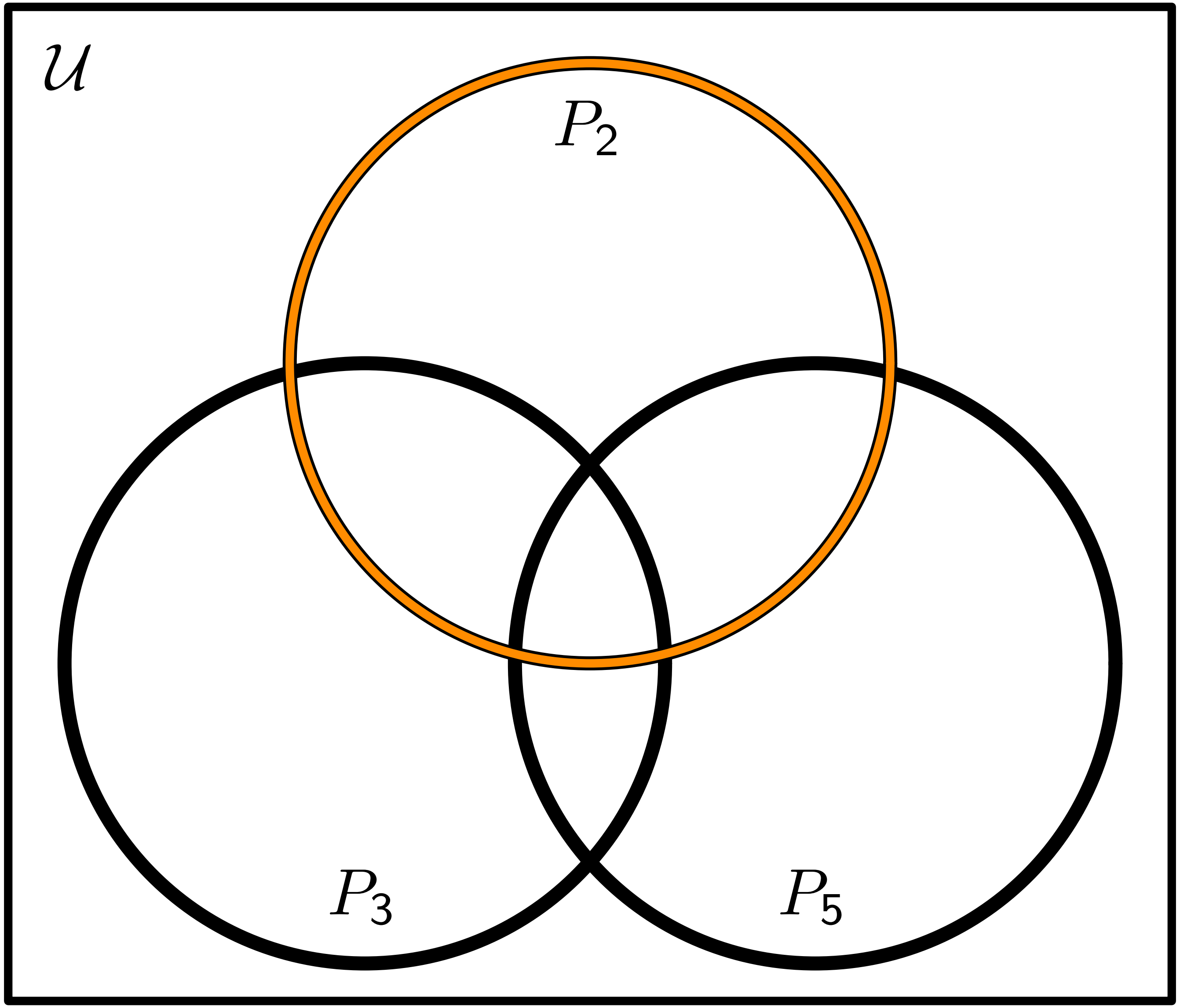
Inclusion–Exclusion



Inclusion–Exclusion

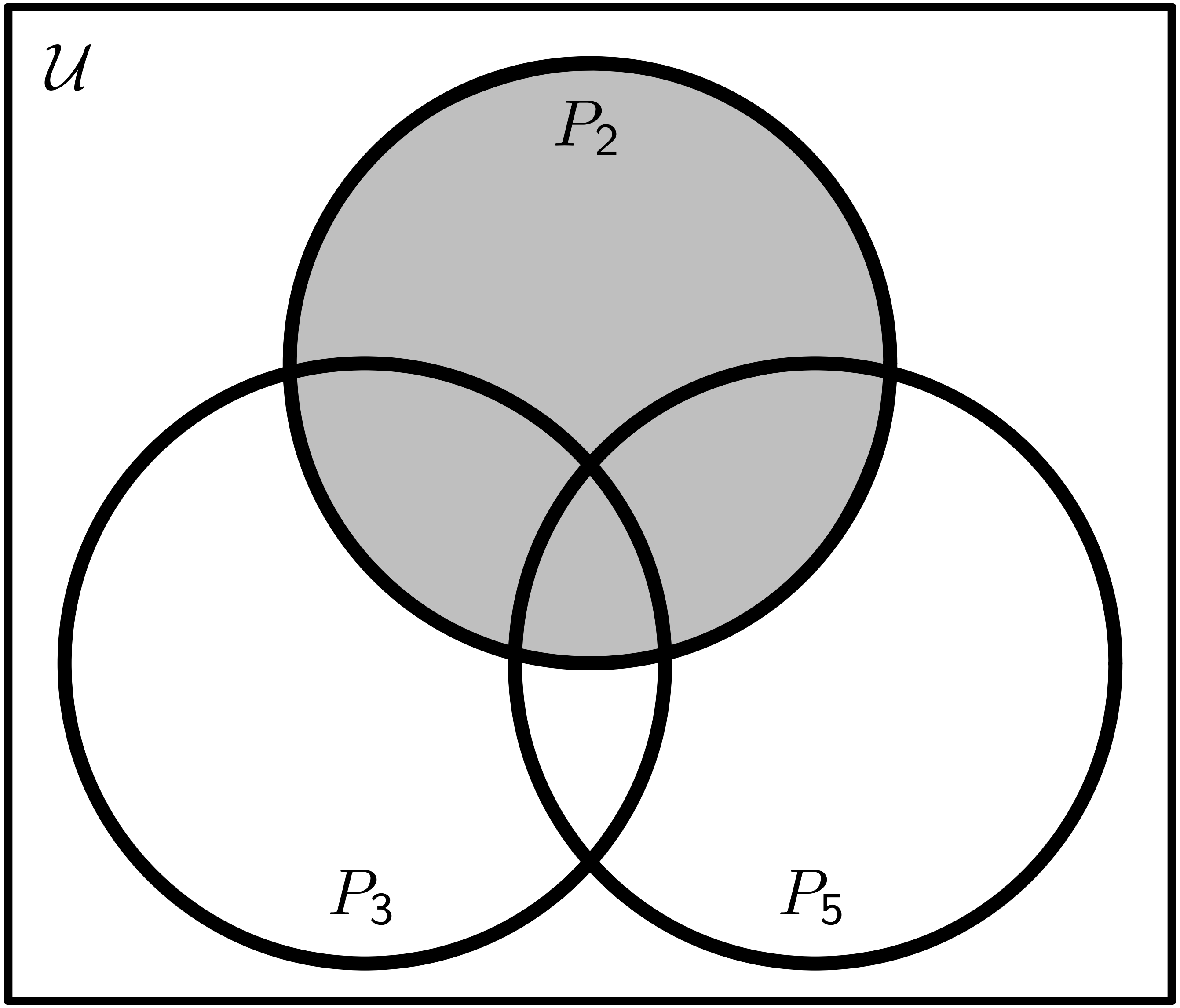


Inclusion–Exclusion



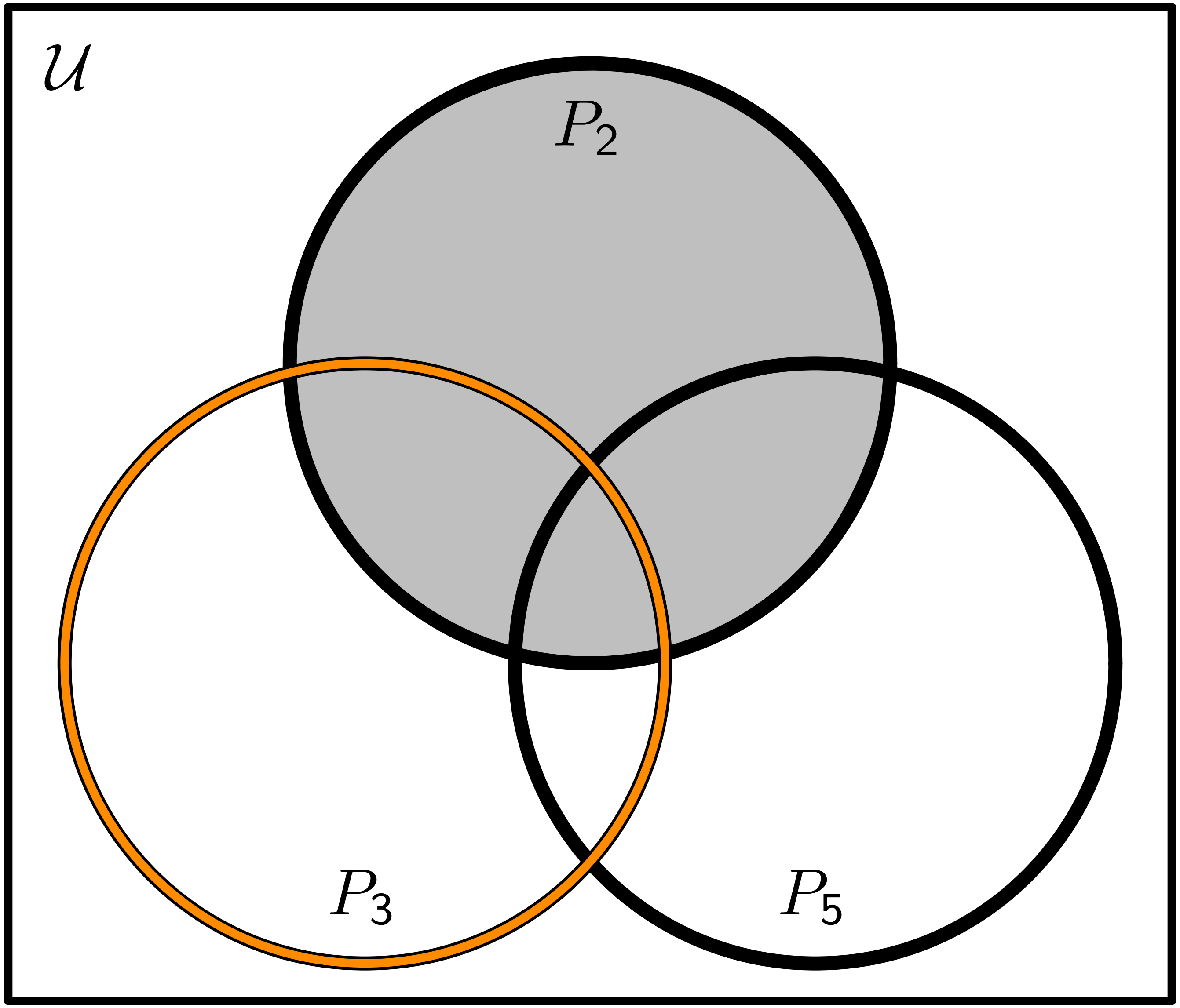
Inclusion–Exclusion

500



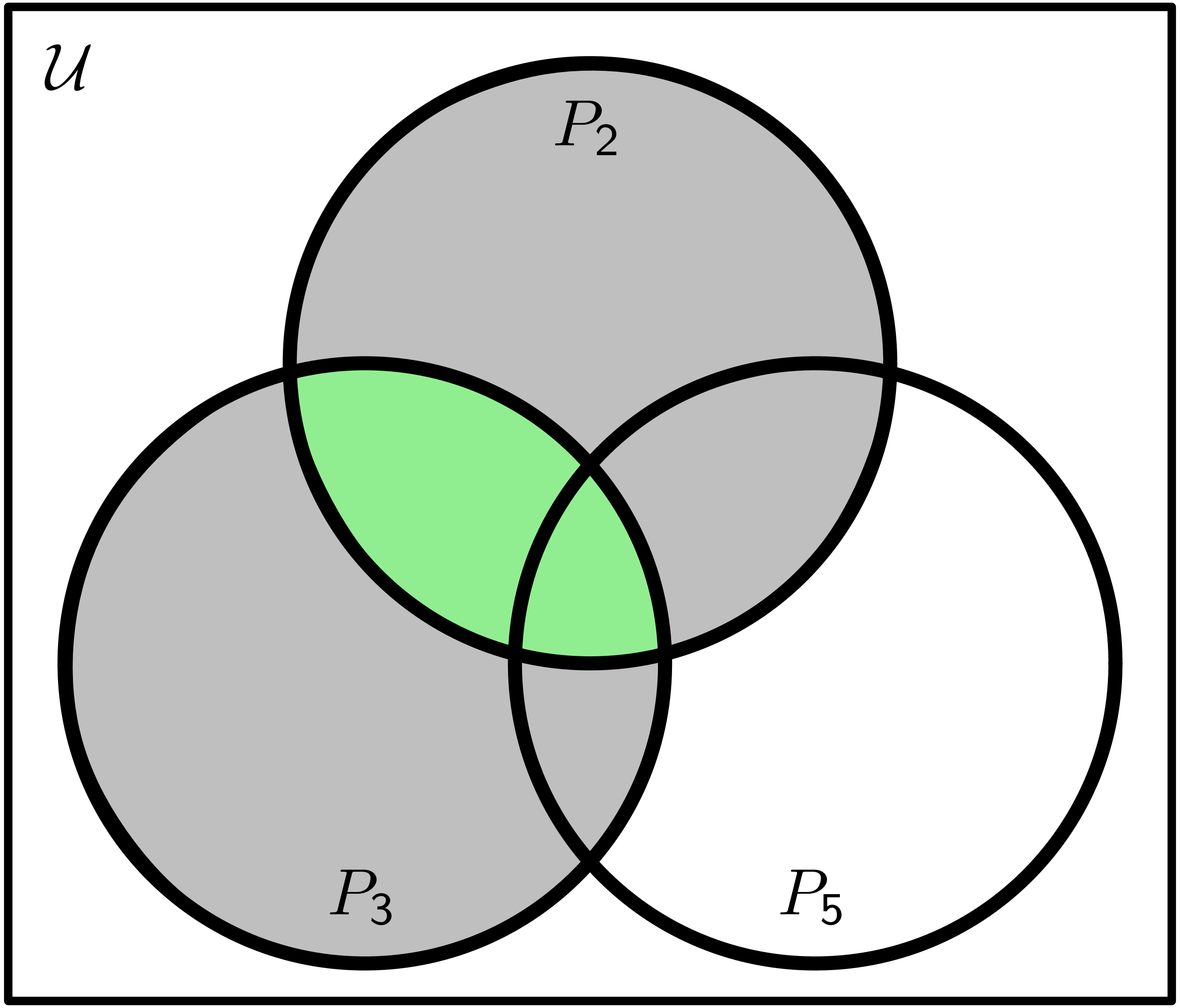
Inclusion–Exclusion

500



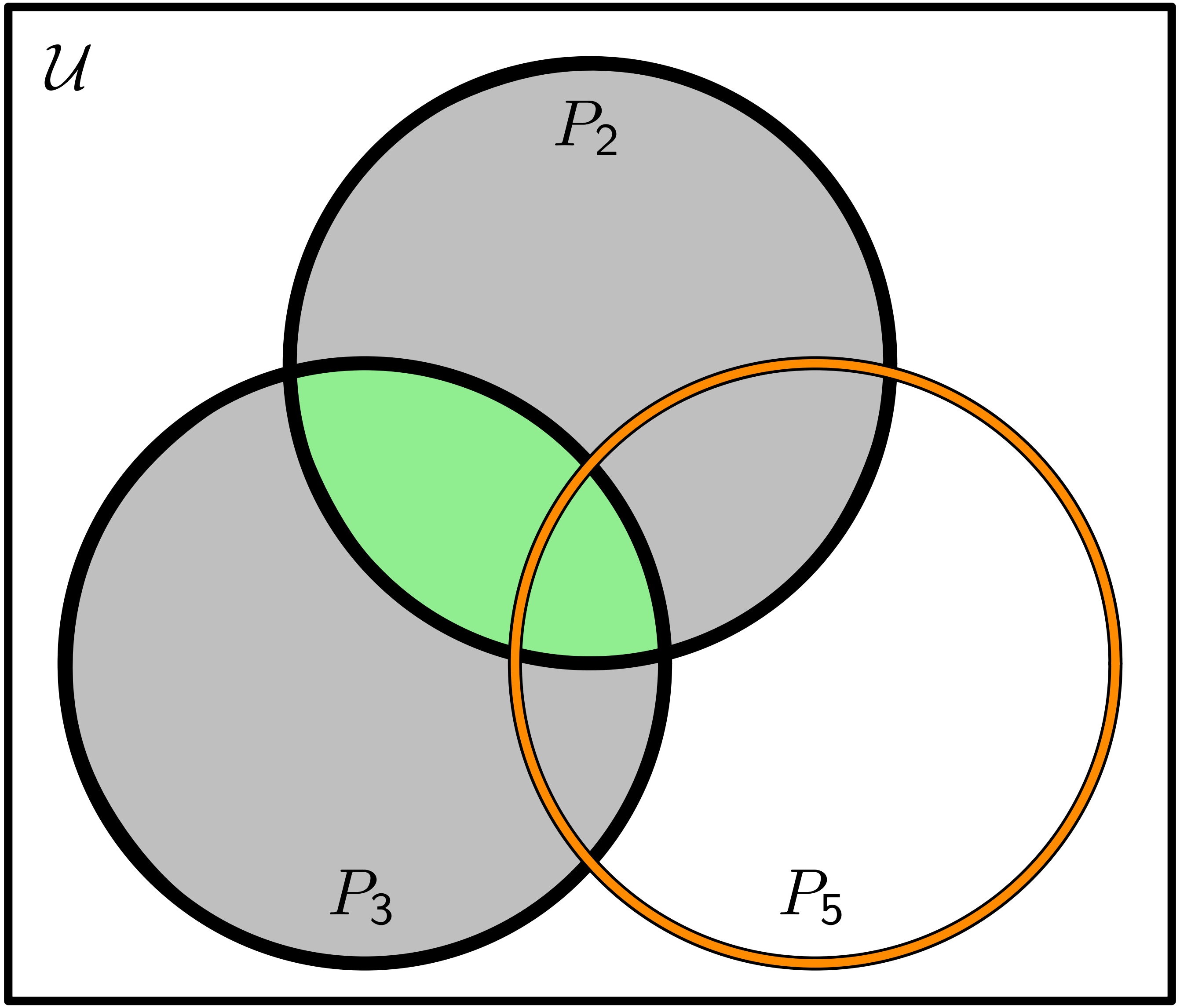
Inclusion–Exclusion

$500 + 333$



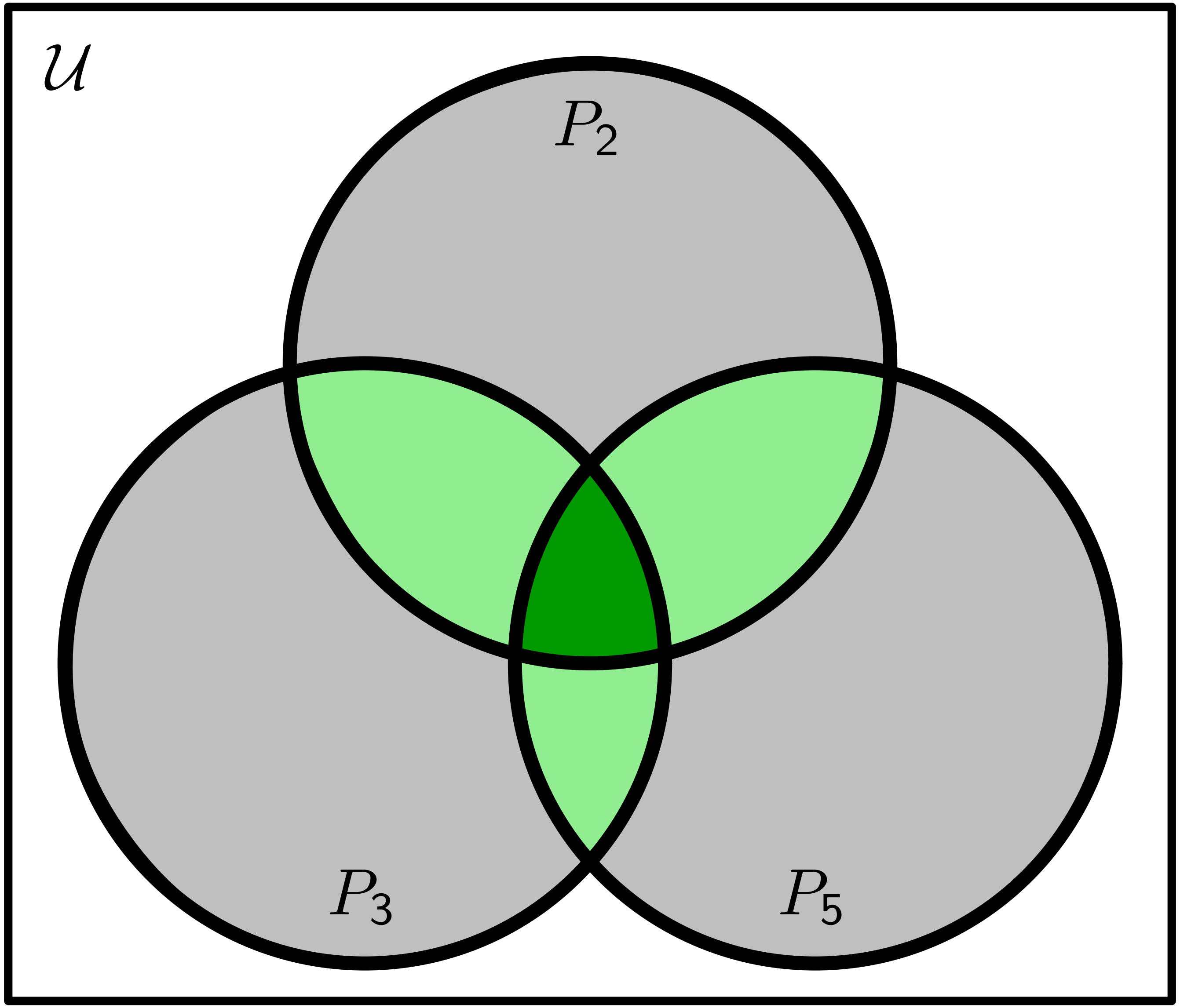
Inclusion–Exclusion

$500 + 333$



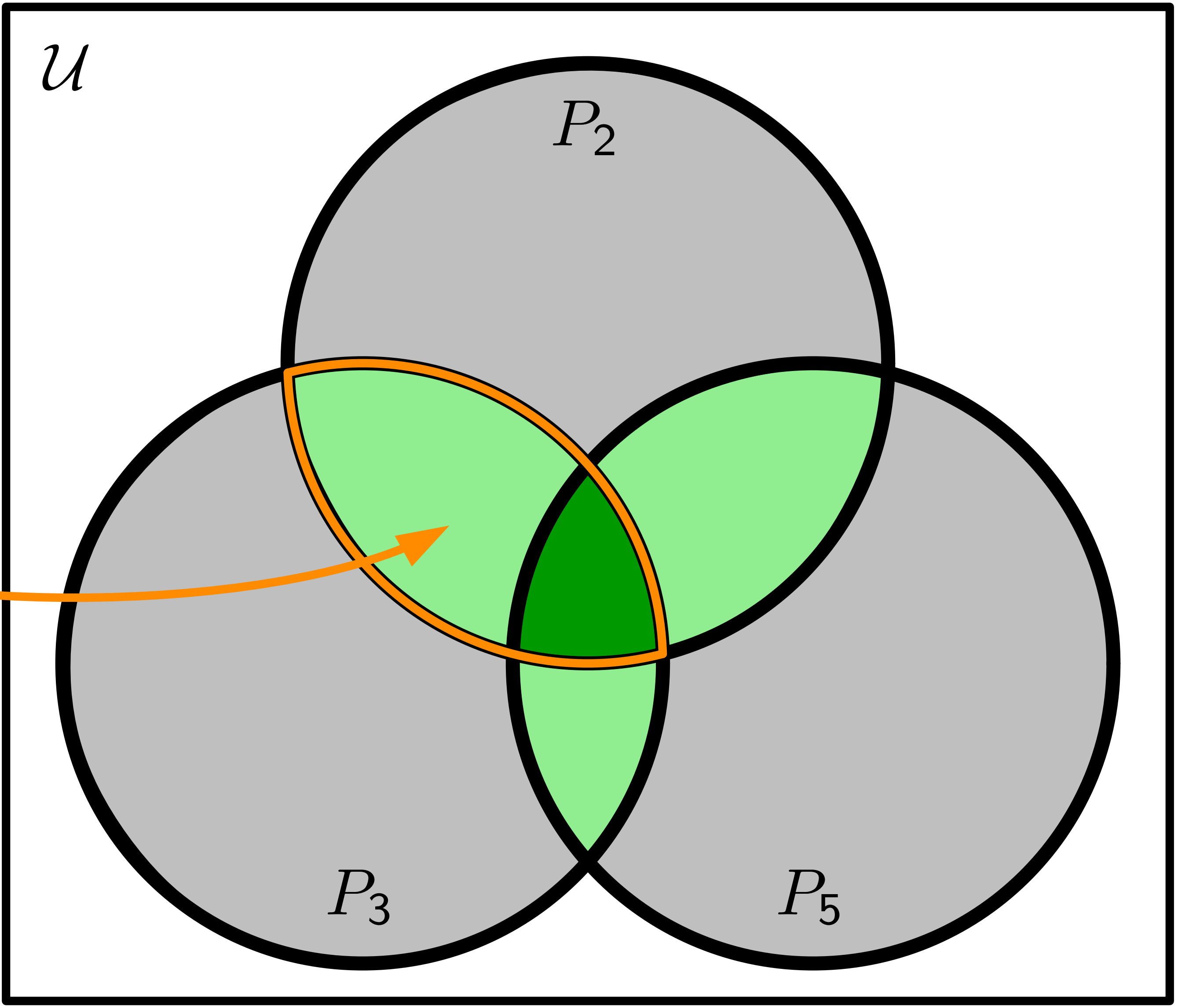
Inclusion–Exclusion

$500 + 333 + 200$



Inclusion–Exclusion

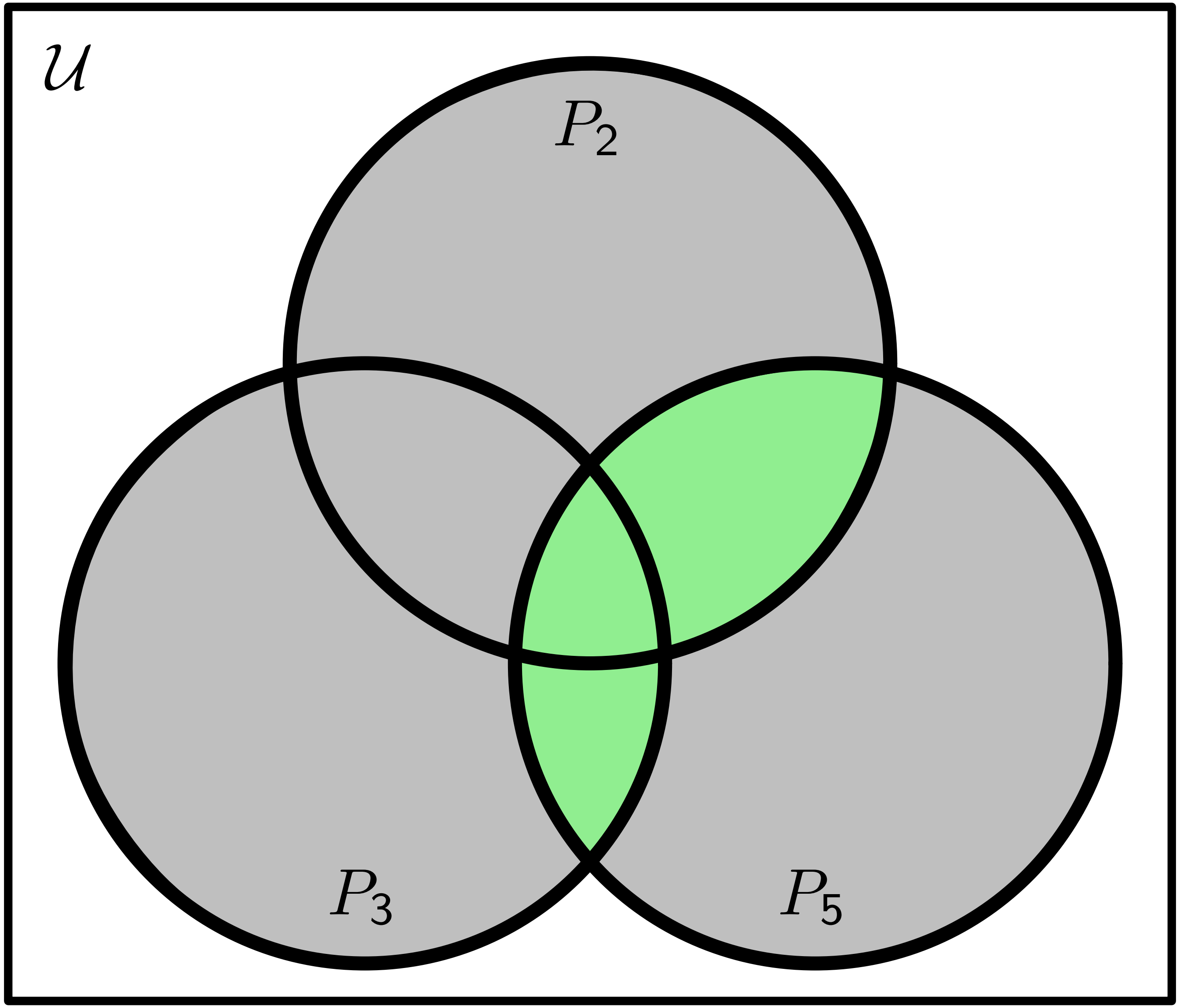
$500 + 333 + 200$



$|P_2 \cap P_3|$

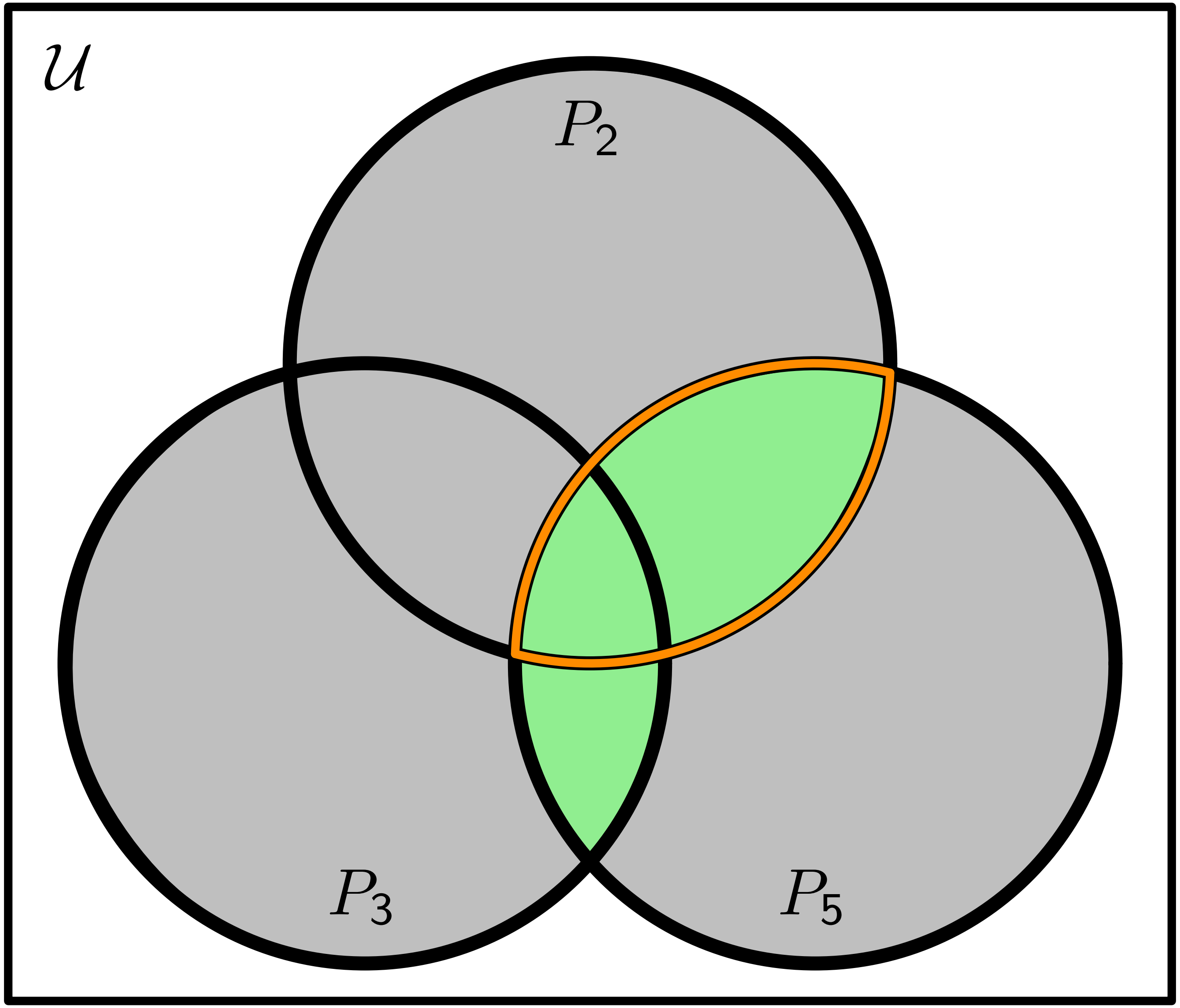
Inclusion–Exclusion

$500 + 333 + 200 - 166$



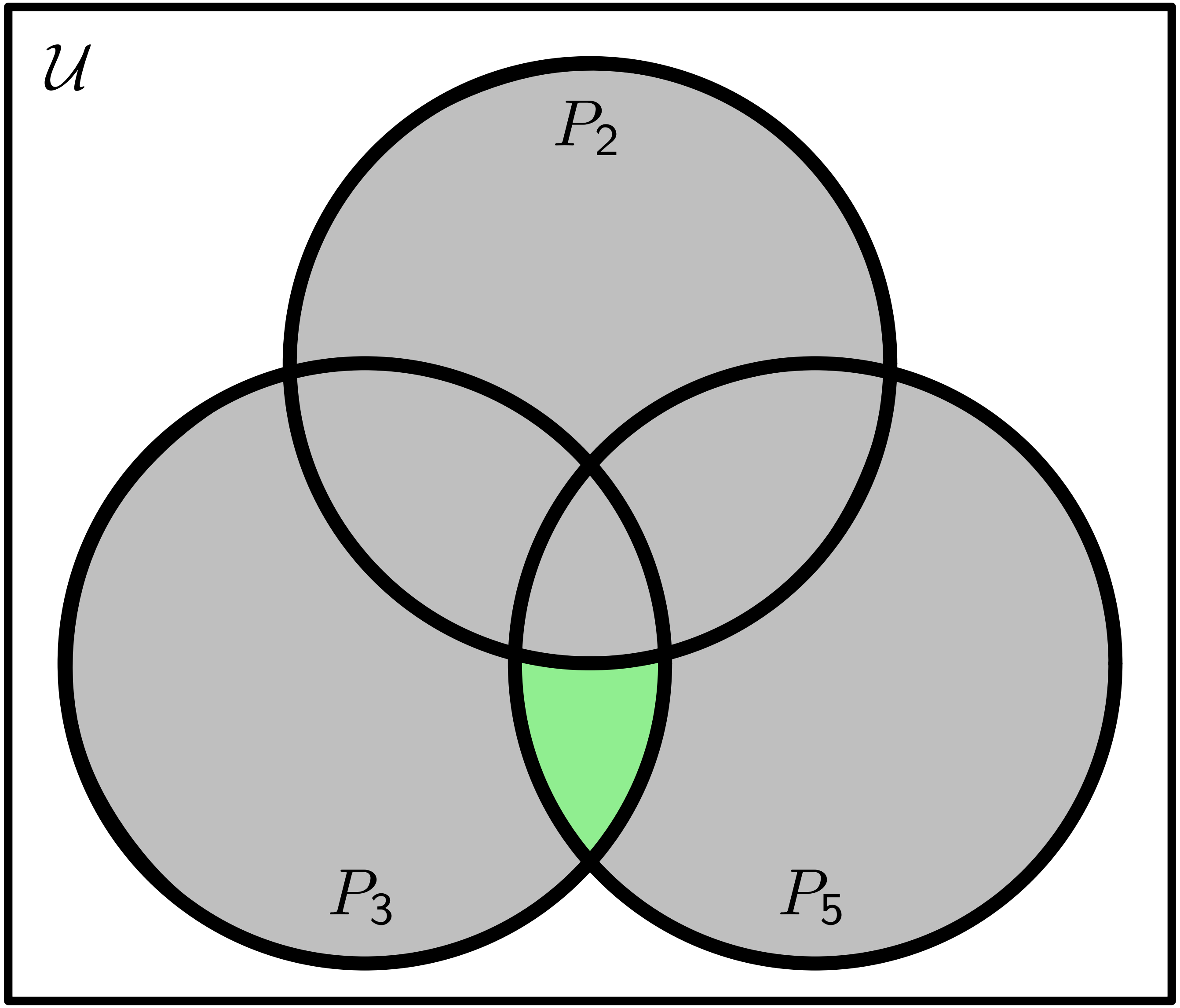
Inclusion–Exclusion

$500 + 333 + 200 - 166$



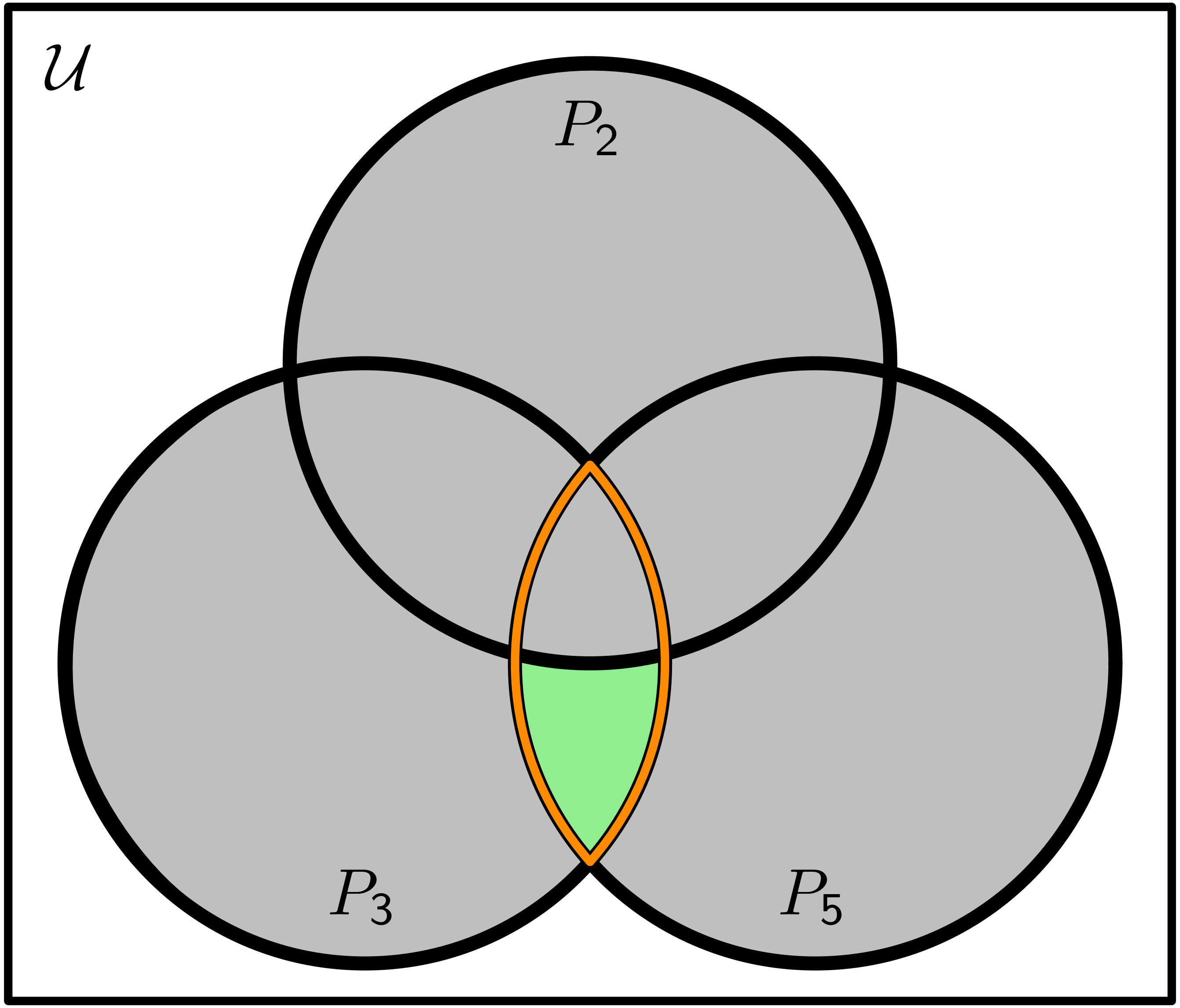
Inclusion–Exclusion

$500 + 333 + 200 - 166 - 100$



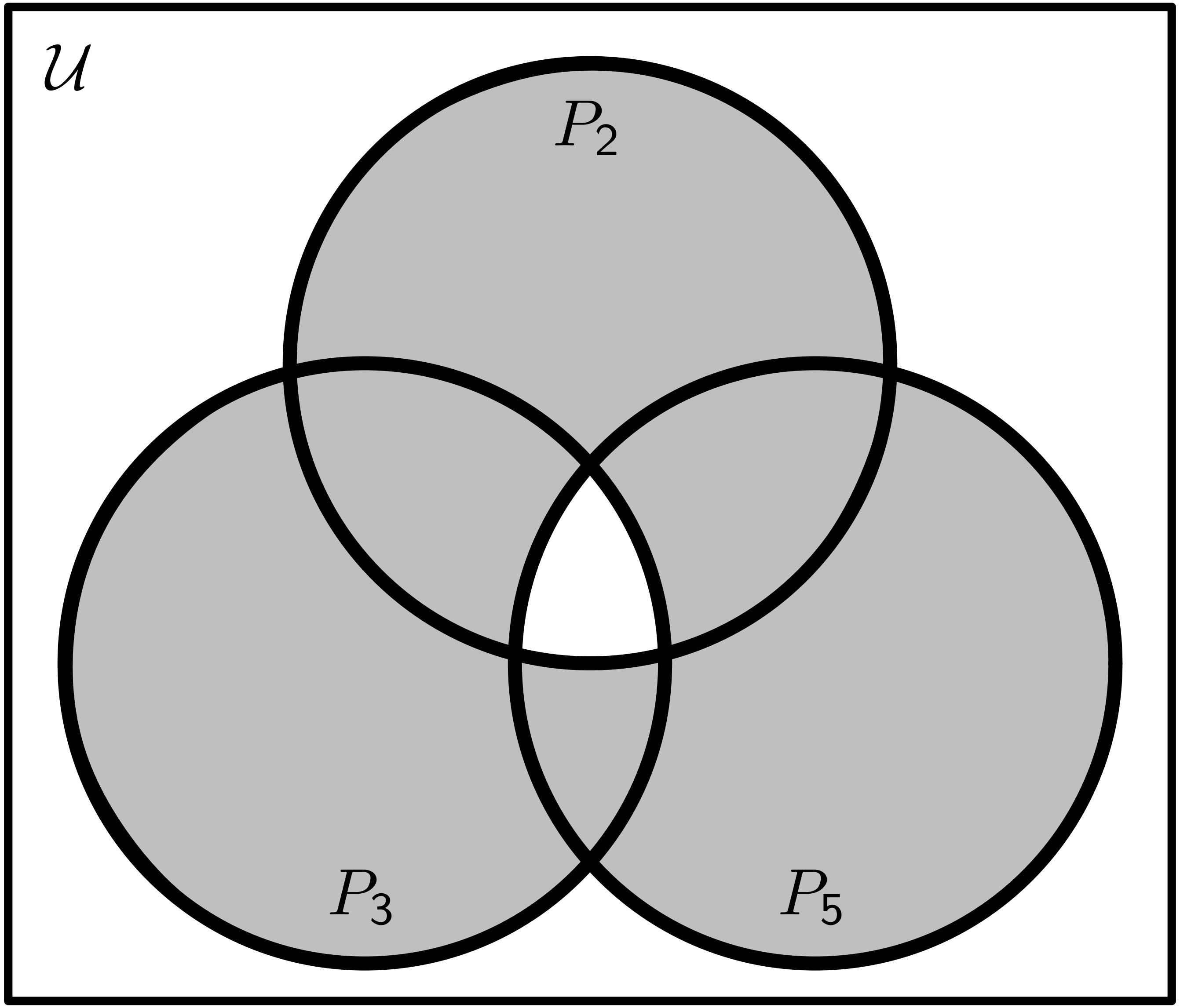
Inclusion–Exclusion

$500 + 333 + 200 - 166 - 100$



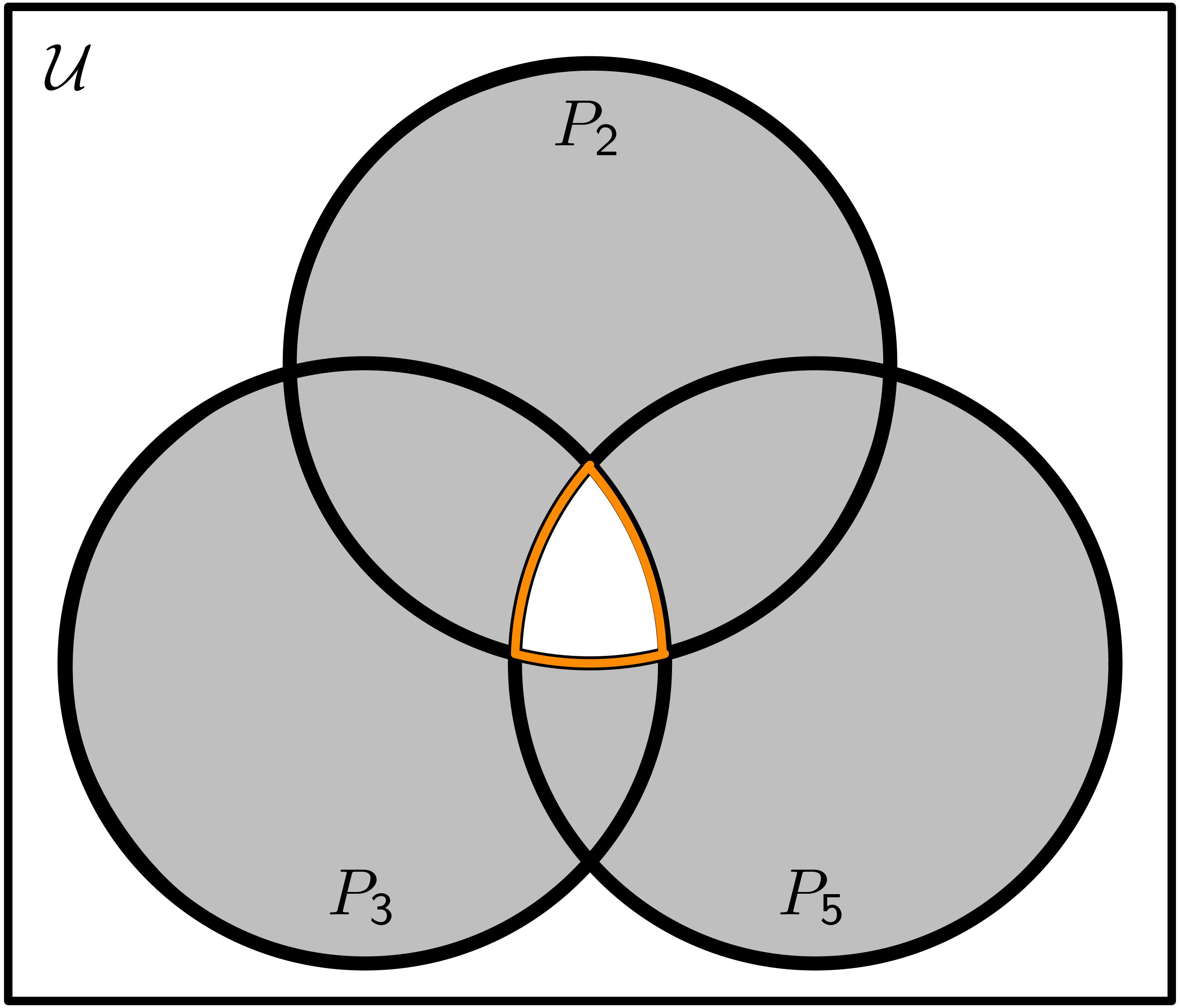
Inclusion–Exclusion

$500 + 333 + 200 - 166 - 100 - 66$



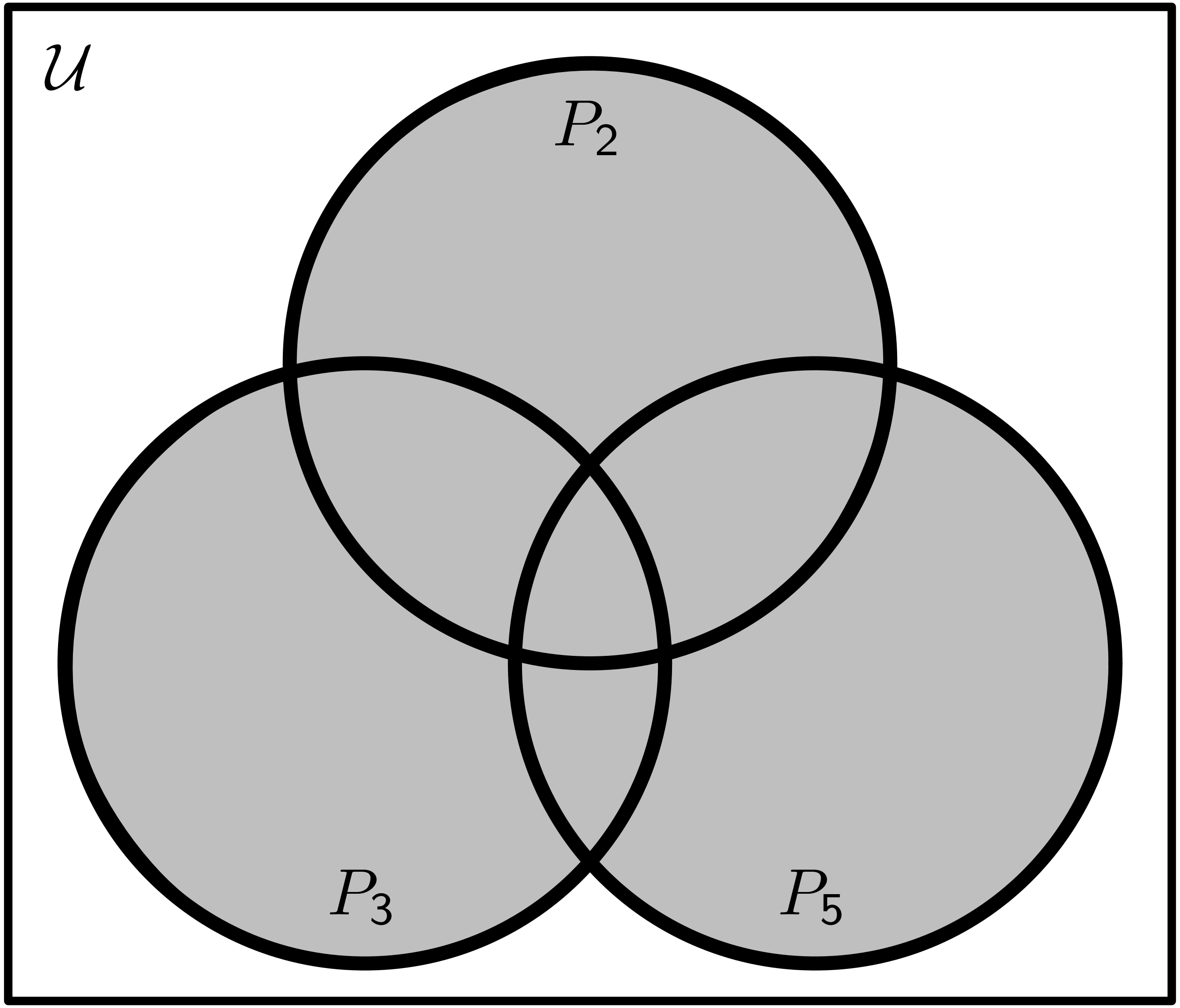
Inclusion–Exclusion

$500 + 333 + 200 - 166 - 100 - 66$



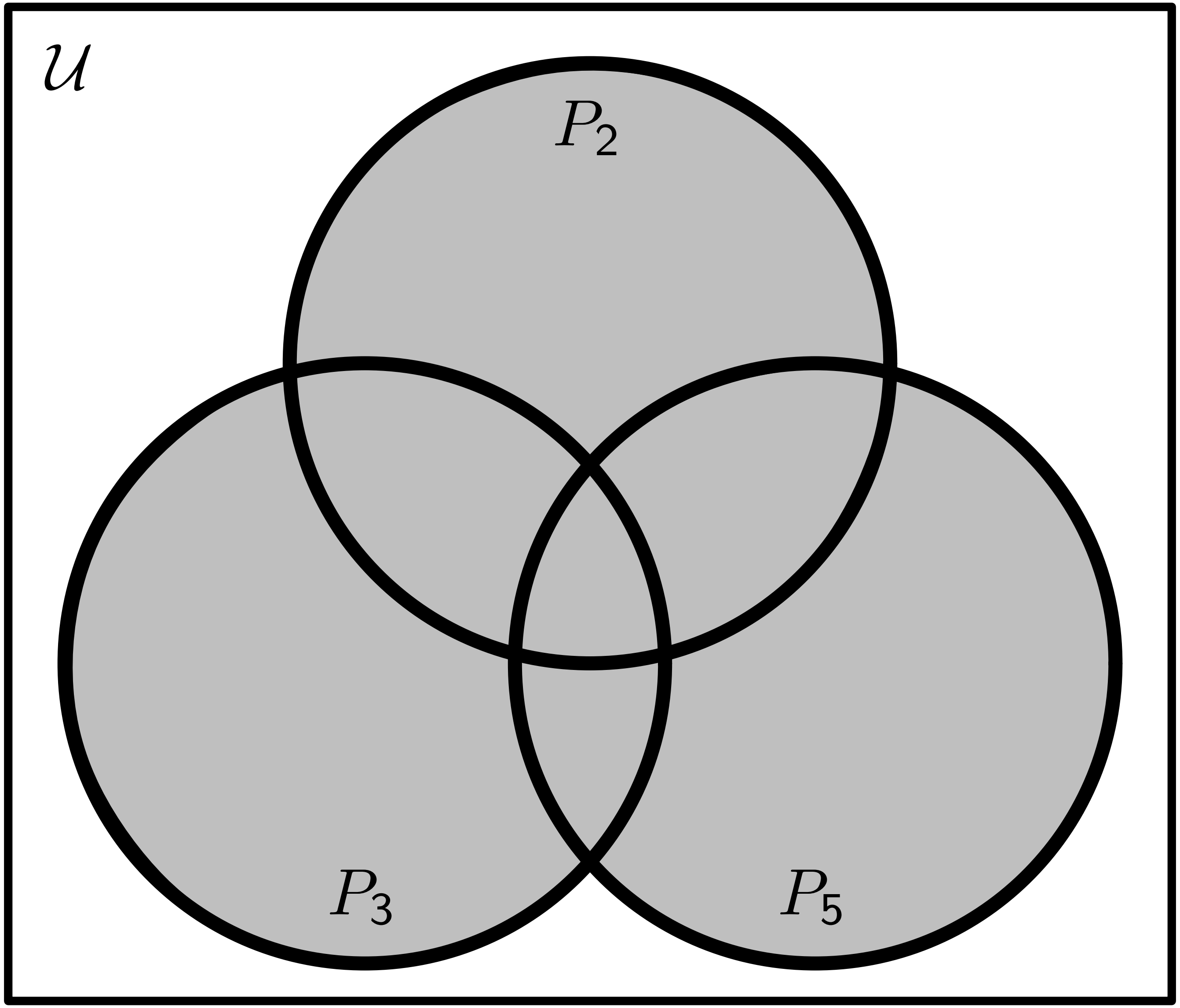
Inclusion–Exclusion

$500 + 333 + 200 - 166 - 100 - 66 + 33$



Inclusion–Exclusion

$$500 + 333 + 200 - 166 - 100 - 66 + 33 \; = \mathbf{734}$$



Inclusion–Exclusion

$$500 + 333 + 200 - 166 - 100 - 66 + 33 = \mathbf{734}$$

$$|P_2 \cup P_3 \cup P_5| =$$

Inclusion–Exclusion

$$500 + 333 + 200 - 166 - 100 - 66 + 33 = \mathbf{734}$$

$$|P_2 \cup P_3 \cup P_5| = + |P_2| + |P_3| + |P_5|$$

Inclusion–Exclusion

$$500 + 333 + 200 - 166 - 100 - 66 + 33 = \mathbf{734}$$

$$\begin{aligned} |P_2 \cup P_3 \cup P_5| = {} & + |P_2| + |P_3| + |P_5| \\ & - |P_2 \cap P_3| - |P_2 \cap P_5| - |P_3 \cap P_5| \end{aligned}$$

Inclusion–Exclusion

$$500 + 333 + 200 - 166 - 100 - 66 + 33 = \mathbf{734}$$

$$\begin{aligned}|P_2 \cup P_3 \cup P_5| = {} & + |P_2| + |P_3| + |P_5| \\ & - |P_2 \cap P_3| - |P_2 \cap P_5| - |P_3 \cap P_5| \\ & + |P_2 \cap P_3 \cap P_5|\end{aligned}$$

Inclusion–Exclusion

$$500 + 333 + 200 - 166 - 100 - 66 + 33 = \mathbf{734}$$

$$\begin{aligned} |P_2 \cup P_3 \cup P_5| = {} & + |P_2| + |P_3| + |P_5| \\ & - |P_2 \cap P_3| - |P_2 \cap P_5| - |P_3 \cap P_5| \\ & + |P_2 \cap P_3 \cap P_5| \end{aligned}$$

$$= \left| \bigcup_{i \in \{2,3,5\}} P_i \right| = \sum_{\varnothing \neq J \subseteq \{2,3,5\}} \qquad \left| \bigcap_{j \in J} P_j \right|$$

Inclusion–Exclusion

$$500 + 333 + 200 - 166 - 100 - 66 + 33 = \mathbf{734}$$

$$\begin{aligned}|P_2 \cup P_3 \cup P_5| = {} & + |P_2| + |P_3| + |P_5| \\ & - |P_2 \cap P_3| - |P_2 \cap P_5| - |P_3 \cap P_5| \\ & + |P_2 \cap P_3 \cap P_5|\end{aligned}$$

$$= \left| \bigcup_{i \in \{2,3,5\}} P_i \right| = \sum_{\varnothing \neq J \subseteq \{2,3,5\}} (-1)^{|J|-1} \left| \bigcap_{j \in J} P_j \right|$$

Inclusion–Exclusion

$$500 + 333 + 200 - 166 - 100 - 66 + 33 = \mathbf{734}$$

$$\begin{aligned}|P_2 \cup P_3 \cup P_5| = {} & + |P_2| + |P_3| + |P_5| \\ & - |P_2 \cap P_3| - |P_2 \cap P_5| - |P_3 \cap P_5| \\ & + |P_2 \cap P_3 \cap P_5|\end{aligned}$$

$$= \left| \bigcup_{i \in \{2,3,5\}} P_i \right| = \sum_{\varnothing \neq J \subseteq \{2,3,5\}} (-1)^{|J|-1} \left| \bigcap_{j \in J} P_j \right|$$

1 × union

7 × intersection

An Inclusion–Exclusion Theorem

Given N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$,

An Inclusion–Exclusion Theorem

Given N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$,
for each $\mathcal{S}$ with $\emptyset \subseteq \mathcal{S} \subseteq \mathcal{P}$, let
$N(\mathcal{S}) := \#$ objects satisfying the properties in $\mathcal{S}$.

An Inclusion–Exclusion Theorem

Given N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$,
for each $\mathcal{S}$ with $\emptyset \subseteq \mathcal{S} \subseteq \mathcal{P}$, let
$N(\mathcal{S}) := \#$ objects satisfying the properties in $\mathcal{S}$.

($N_0 := \#$ objects with *no* property from $\mathcal{P}$.)

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

An Inclusion–Exclusion Theorem

Given N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$,
for each $\mathcal{S}$ with $\emptyset \subseteq \mathcal{S} \subseteq \mathcal{P}$, let
$N(\mathcal{S}) :=$ # objects satisfying the properties in $\mathcal{S}$.

($N_0 :=$ # objects with *no* property from $\mathcal{P}$.)

Thm. $N_0 \; = \; \sum_{\mathcal{S}\subseteq\mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

1 union

2^n subsets

An Inclusion–Exclusion Theorem

Given N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$,
for each $\mathcal{S}$ with $\emptyset \subseteq \mathcal{S} \subseteq \mathcal{P}$, let
$N(\mathcal{S}) :=$ # objects satisfying the properties in $\mathcal{S}$.

($N_0 :=$ # objects with *no* property from $\mathcal{P}$.)

Thm. $N_0 = \sum_{\mathcal{S}\subseteq\mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

1 union

2^n subsets

Is this useful?

An Inclusion–Exclusion Theorem

Given N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$,
for each $\mathcal{S}$ with $\emptyset \subseteq \mathcal{S} \subseteq \mathcal{P}$, let
$N(\mathcal{S}) :=$ # objects satisfying the properties in $\mathcal{S}$.

($N_0 :=$ # objects with *no* property from $\mathcal{P}$.)

Thm. $N_0 \;=\; \sum_{\mathcal{S}\subseteq\mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

1 union

2^n subsets

Is this useful?

“One of the most useful principles of enumeration in discrete probability and combinatorial theory is the celebrated principle of inclusion–exclusion. When skillfully applied, this principle has yielded the solution to many a combinatorial problem.”

Gian-Carlo Rota [1932–1999]

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof.

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} =$$

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} = \sum_{i=0}^{s} \qquad (-1)^i =$$

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} = \sum_{i=0}^{s} \binom{s}{i} (-1)^i =$$

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} = \sum_{i=0}^{s} \binom{s}{i} (-1)^i = \mathbf{0}, \text{ because:}$$

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} = \sum_{i=0}^{s} \binom{s}{i} (-1)^i = \mathbf{0}, \text{ because:}$$

– if s is odd: $\binom{s}{0} - \binom{s}{1} + \binom{s}{2} \mp \cdots - \binom{s}{s-2} + \binom{s}{s-1} - \binom{s}{s}$

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} = \sum_{i=0}^{s} \binom{s}{i} (-1)^i = \mathbf{0}, \text{ because:}$$

– if s is odd: $\binom{s}{0} - \binom{s}{1} + \binom{s}{2} \mp \cdots - \binom{s}{s-2} + \binom{s}{s-1} - \binom{s}{s}$

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} = \sum_{i=0}^{s} \binom{s}{i} (-1)^i = \mathbf{0}, \text{ because:}$$

– if s is odd: $\binom{s}{0} - \binom{s}{1} + \binom{s}{2} \mp \cdots - \binom{s}{s-2} + \binom{s}{s-1} - \binom{s}{s}$

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

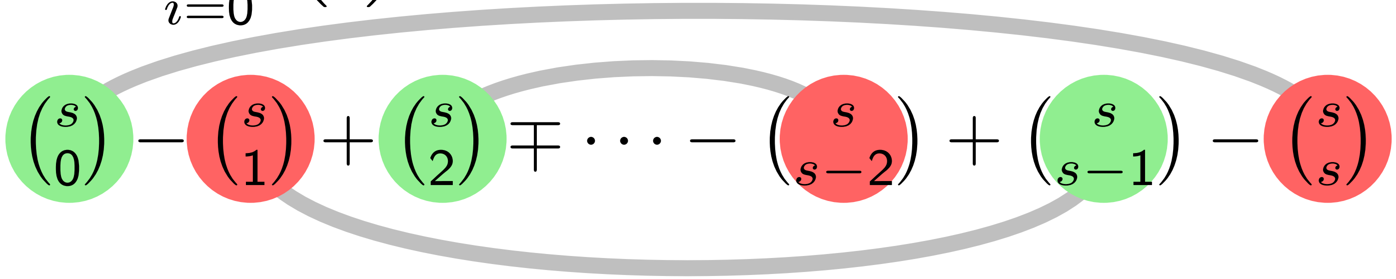
Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \ldots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} = \sum_{i=0}^{s} \binom{s}{i} (-1)^i = \mathbf{0}, \text{ because:}$$



– if s is odd: $\binom{s}{0} - \binom{s}{1} + \binom{s}{2} \mp \cdots - \binom{s}{s-2} + \binom{s}{s-1} - \binom{s}{s}$

Proof

Thm. $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$.

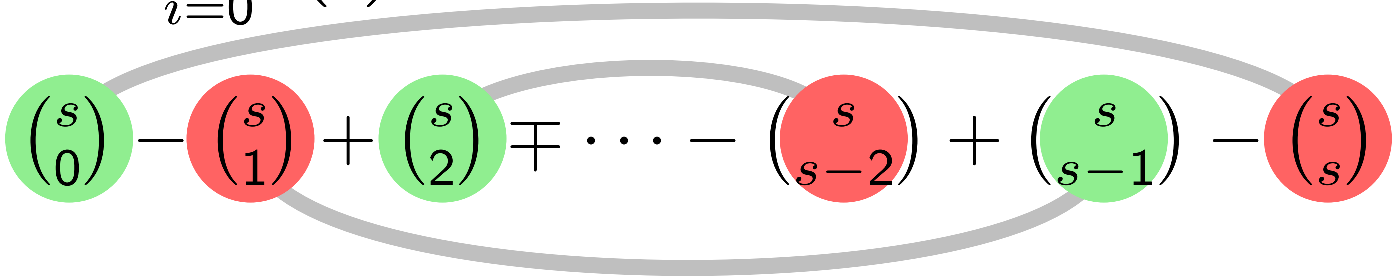
Proof. Property-less objects are counted exactly once on both sides (on the right side when $\mathcal{S} = \emptyset$).

Consider an object with *precisely* properties $\mathcal{S} = \{P_{i_1}, \dots, P_{i_s}\}$, $s \geq 1$.

We count this object in $N(\mathcal{S}')$ precisely when $\mathcal{S}' \subseteq \mathcal{S}$.

For an object with properties $\mathcal{S}$, we observe:

$$\sum_{\mathcal{S}' \subseteq \mathcal{S}} (-1)^{|\mathcal{S}'|} = \sum_{i=0}^{s} \binom{s}{i} (-1)^i = \mathbf{0}, \text{ because:}$$



– if s is odd: $\binom{s}{0} - \binom{s}{1} + \binom{s}{2} \mp \cdots - \binom{s}{s-2} + \binom{s}{s-1} - \binom{s}{s}$

– if s is even – **exercise!** □

Corollary

Recall: N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$.

Cor. Let $\bar{N}(\mathcal{S})$ be the number of objects that have *none* of the properties in $\mathcal{S}$. Then

$$N(\mathcal{P}) = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} \bar{N}(\mathcal{S})$$

Corollary

Recall: N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$.

Cor. Let $\bar{N}(\mathcal{S})$ be the number of objects that have *none* of the properties in $\mathcal{S}$. Then

$$N(\mathcal{P}) = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} \bar{N}(\mathcal{S})$$

Proof. Set $P_i{}' :=$ "object does not have property P_i".

Corollary

Recall: N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$.

Cor. Let $\bar{N}(\mathcal{S})$ be the number of objects that have *none* of the properties in $\mathcal{S}$. Then

$$N(\mathcal{P}) = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} \bar{N}(\mathcal{S})$$

Proof. Set $P_i' :=$ "object does not have property P_i".

Apply **Thm.** $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$ to:

Corollary

Recall: N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$.

Cor. Let $\bar{N}(\mathcal{S})$ be the number of objects that have *none* of the properties in $\mathcal{S}$. Then

$$N(\mathcal{P}) = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} \bar{N}(\mathcal{S})$$

Proof. Set $P_i{}' :=$ "object does not have property P_i".

Apply **Thm.** $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$ to:

$N_0' =$

Corollary

Recall: N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$.

Cor. Let $\bar{N}(\mathcal{S})$ be the number of objects that have *none* of the properties in $\mathcal{S}$. Then

$$N(\mathcal{P}) = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} \bar{N}(\mathcal{S})$$

Proof. Set $P_i{}' :=$ “object does not have property P_i”.

Apply **Thm.** $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$ to:

$$N_0' = N(\mathcal{P})$$

Corollary

Recall: N objects and n properties $\mathcal{P} = \{P_1, \dots, P_n\}$.

Cor. Let $\bar{N}(\mathcal{S})$ be the number of objects that have *none* of the properties in $\mathcal{S}$. Then

$$N(\mathcal{P}) = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} \bar{N}(\mathcal{S})$$

Proof. Set $P_i{}' :=$ "object does not have property P_i".

Apply **Thm.** $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$ to:

$N_0' = N(\mathcal{P})$

$N'(P_{i_1}', \dots, P_{i_s}') =$

Corollary

Recall: N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$.

Cor. Let $\bar{N}(\mathcal{S})$ be the number of objects that have *none* of the properties in $\mathcal{S}$. Then

$$N(\mathcal{P}) = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} \bar{N}(\mathcal{S})$$

Proof. Set $P_i{}' :=$ "object does not have property P_i".

Apply **Thm.** $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$ to:

$$N_0' = N(\mathcal{P})$$

$$N'(P_{i_1}', \ldots, P_{i_s}') = \bar{N}(P_{i_1}, \ldots, P_{i_s}) \qquad \square$$

Corollary

Recall: N objects and n properties $\mathcal{P} = \{P_1, \ldots, P_n\}$.

Cor. Let $\bar{N}(\mathcal{S})$ be the number of objects that have *none* of the properties in $\mathcal{S}$. Then

$$N(\mathcal{P}) = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} \bar{N}(\mathcal{S})$$

But is it useful?

Proof. Set $P_i' :=$ "object does not have property P_i".

Apply **Thm.** $N_0 = \sum_{\mathcal{S} \subseteq \mathcal{P}} (-1)^{|\mathcal{S}|} N(\mathcal{S})$ to:

$$N_0' = N(\mathcal{P})$$

$$N'(P_{i_1}', \ldots, P_{i_s}') = \bar{N}(P_{i_1}, \ldots, P_{i_s})$$

□

Directed Hamiltonian Path

Given: Directed Graph $G = (V, E)$,
two special vertices $s, t \in V$.

Question: Is there a *Hamiltonian path* from s to t,
i.e., an s-t-path spanning the vertices of G?

Directed Hamiltonian Path

Given: Directed Graph $G = (V, E)$,
two special vertices $s, t \in V$.

Question: Is there a *Hamiltonian path* from s to t,
i.e., an s-t-path spanning the vertices of G?

Brute Force?

Directed Hamiltonian Path

Given: Directed Graph $G = (V, E)$,
two special vertices $s, t \in V$.

Question: Is there a *Hamiltonian path* from s to t,
i.e., an s-t-path spanning the vertices of G?

Brute Force? Try all permutations of $V \rightsquigarrow \Theta(n!) = 2^{\Theta(n \log n)}$

Directed Hamiltonian Path

Given: Directed Graph $G = (V, E)$,
two special vertices $s, t \in V$.

Question: Is there a *Hamiltonian path* from s to t,
i.e., an s-t-path spanning the vertices of G?

Brute Force? Try all permutations of $V \rightsquigarrow \Theta(n!) = 2^{\Theta(n \log n)}$

Another idea:

Directed Hamiltonian Path

Given: Directed Graph $G = (V, E)$,
two special vertices $s, t \in V$.

Question: Is there a *Hamiltonian path* from s to t,
i.e., an s-t-path spanning the vertices of G?

Brute Force? Try all permutations of $V \rightsquigarrow \Theta(n!) = 2^{\Theta(n \log n)}$

Another idea: Via TSP $\Rightarrow$

Directed Hamiltonian Path

Given: Directed Graph $G = (V, E)$,
two special vertices $s, t \in V$.

Question: Is there a *Hamiltonian path* from s to t,
i.e., an s-t-path spanning the vertices of G?

Brute Force? Try all permutations of $V \rightsquigarrow \Theta(n!) = 2^{\Theta(n \log n)}$

Another idea: Via TSP $\Rightarrow$ $O^*(2^n)$ time and *space*.

A Better Algorithm for Hamiltonian Path

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple.

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s, t\}$

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s, t\}$

Property $v \in V'$: s–t walk visits v.

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s,t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s,t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) :=$

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s,t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) := \#$ s–t walks of length $n-1$

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s,t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) := \#$ s–t walks of length $n-1$, *avoiding* W.

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s, t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) :=$ # s–t walks of length $n-1$, *avoiding* W.

$N(V') :=$

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s,t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) :=$ # s–t walks of length $n-1$, *avoiding* W.

$N(V') :=$ # directed Hamiltonian s–t paths.

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s, t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) :=$ # s–t walks of length $n-1$, *avoiding* W.

$N(V') :=$ # directed Hamiltonian s–t paths.

By inclusion–exclusion:

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s, t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) :=$ # s–t walks of length $n-1$, *avoiding* W.

$N(V') :=$ # directed Hamiltonian s–t paths.

By inclusion–exclusion:

$$N(V') =$$

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s, t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) := \#$ s–t walks of length $n-1$, *avoiding* W.

$N(V') := \#$ directed Hamiltonian s–t paths.

By inclusion–exclusion:

$$N(V') = \sum_{W \subseteq V'} (-1)^{|W|} \bar{N}(W)$$

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s,t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) :=$ # s–t walks of length $n-1$, *avoiding* W.

$N(V') :=$ # directed Hamiltonian s–t paths.

By inclusion–exclusion:

$$N(V') = \sum_{W \subseteq V'} (-1)^{|W|} \bar{N}(W)$$

Compute $\bar{N}(W)$ for each $W \subseteq V'$ separately.

A Better Algorithm for Hamiltonian Path

Thm. The *number* of directed Hamiltonian paths from s to t can be determined in $O^*(2^n)$ time using only *polynomial space*. [Karp '82]

Proof. Objects: Directed s–t walks of length $n-1$, not necessarily simple. $V' := V - \{s,t\}$

Property $v \in V'$: s–t walk visits v. For $W \subseteq V'$:

$\bar{N}(W) :=$ # s–t walks of length $n-1$, *avoiding* W.

$N(V') :=$ # directed Hamiltonian s–t paths.

By inclusion–exclusion:

$$N(V') = \sum_{W \subseteq V'} (-1)^{|W|} \bar{N}(W)$$

Compute $\bar{N}(W)$ for each $W \subseteq V'$ separately. How?

Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k, \text{ avoiding } W$$

Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k\text{, avoiding } W$$

$$\bar{N}(W) :=$$

Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k\text{, avoiding } W$$

$$\bar{\mathrm{N}}(W) := P_W[t, n-1]$$

Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k\text{, avoiding } W$$

$$\bar{N}(W) := P_W[t, n-1]$$

Recurrence for the DP:

Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k\text{, avoiding } W$$

$$\bar{N}(W) := P_W[t, n-1]$$

Recurrence for the DP:

- for $k = 0$

$$P_W[u,0] = \left\{ \begin{array}{l} \\ \\ \end{array} \right.$$

Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k \text{, avoiding } W$$

$$\bar{N}(W) := P_W[t, n-1]$$

Recurrence for the DP:

- for $k = 0$

$$P_W[u,0] = \begin{cases} 0 & \text{when } u \neq s \\ 1 & \text{when } u = s \end{cases}$$

Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k, \text{ avoiding } W$$

$$\bar{N}(W) := P_W[t, n-1]$$

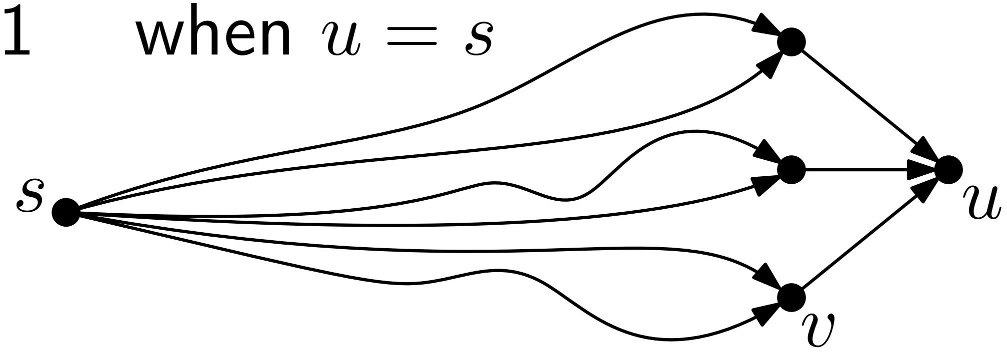
Recurrence for the DP:

- for $k = 0$

$$P_W[u,0] = \begin{cases} 0 & \text{when } u \neq s \\ 1 & \text{when } u = s \end{cases}$$

- for $k \geq 1$

$$P_W[u,k] =$$



Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k\text{, avoiding } W$$

$$\bar{N}(W) := P_W[t, n-1]$$

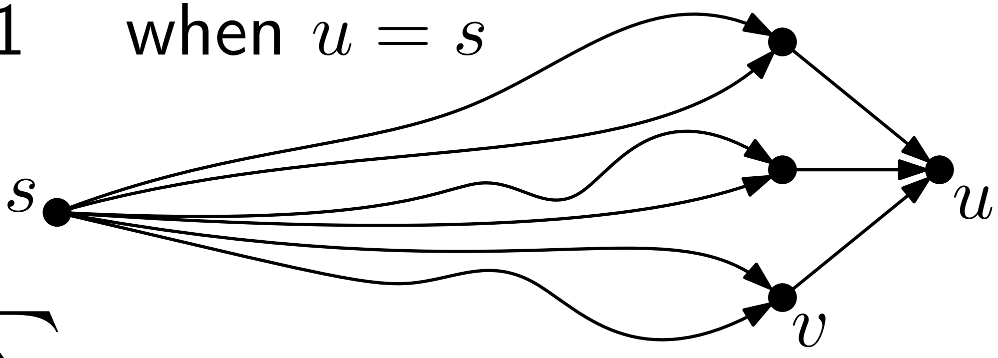
Recurrence for the DP:

- for $k = 0$

$$P_W[u,0] = \begin{cases} 0 & \text{when } u \neq s \\ 1 & \text{when } u = s \end{cases}$$

- for $k \geq 1$

$$P_W[u,k] = \sum_{\substack{vu \in E \\ v \notin W}}$$



Dynamic Program

For $W \subseteq V'$, $k = 0, \ldots, n-1$, and $u \in V \setminus W$, set:

$$P_W[u,k] := \#\ s\text{–}u \text{ walks of length } k \text{, avoiding } W$$

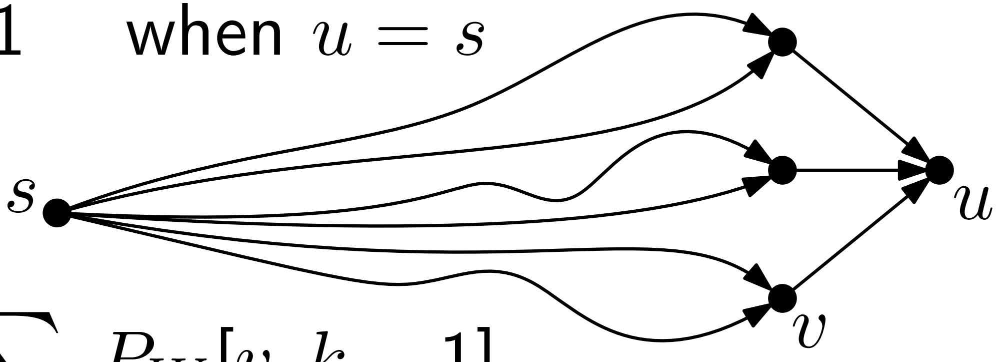
$$\bar{N}(W) := P_W[t, n-1]$$

Recurrence for the DP:

- for $k = 0$

$$P_W[u,0] = \begin{cases} 0 & \text{when } u \neq s \\ 1 & \text{when } u = s \end{cases}$$

- for $k \geq 1$



$$P_W[u,k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v, k-1]$$

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    foreach $W \subseteq V'$ do
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|}\bar{N}(W)$
    return $N(V')$
```

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    foreach $W \subseteq V'$ do
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|}\bar{N}(W)$
    return $N(V')$
```

Runtime:

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    **foreach** $W \subseteq V'$ **do**
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|}\bar{N}(W)$
    **return** $N(V')$
```

Runtime:

$$P_W[u,k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v, k-1]$$

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    **foreach** $W \subseteq V'$ **do**
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|}\bar{N}(W)$
    **return** $N(V')$
```

fix!

Runtime:

$$P_W[u,k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v,k-1]$$

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    foreach $W \subseteq V'$ do
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|} \bar{N}(W)$
    return $N(V')$
```

$\Big\{ O(m^2 n)$ time

[Note that numbers in P_W can be exponential in m!]

fix!

Runtime:

$$P_W[u, k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v, k-1]$$

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    foreach $W \subseteq V'$ do
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|}\bar{N}(W)$
    return $N(V')$
```

$\left\{\begin{array}{l} O(m^2 n) \\ \text{time} \end{array}\right.$

[Note that numbers in P_W can be exponential in m!]

fix!

Runtime: $O(m^2 n \cdot 2^n)$

$$P_W[u,k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v, k-1]$$

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    foreach $W \subseteq V'$ do
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|} \bar{N}(W)$
    return $N(V')$
```

$\left\{ \begin{array}{l} O(m^2 n) \\ \text{time} \end{array} \right.$

[Note that numbers in P_W can be exponential in $m!$]

fix!

$$P_W[u,k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v, k-1]$$

Runtime: $O(m^2 n \cdot 2^n)$

Space:

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    foreach $W \subseteq V'$ do
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|} \bar{N}(W)$
    return $N(V')$
```

$\left\{ \begin{array}{l} O(m^2 n) \\ \text{time} \end{array} \right.$

[Note that numbers in P_W can be exponential in $m!$]

fix!

$$P_W[u, k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v, k-1]$$

Runtime: $O(m^2 n \cdot 2^n)$

Space: $O(mn)$ – if we reuse the memory used by the DP in each iteration of the loop.

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    foreach $W \subseteq V'$ do
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|} \bar{N}(W)$
    return $N(V')$
```

$\left\{ \begin{array}{l} O(m^2 n) \\ \text{time} \end{array} \right.$

[Note that numbers in P_W can be exponential in $m!$]

fix!

$$P_W[u, k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v, k-1]$$

Runtime: $O(m^2 n \cdot 2^n)$

Space: $O(mn)$ – if we reuse the memory used by the DP in each iteration of the loop.

How can we use this to *find* a Hamiltonian Path?

Algorithmic Application of the IE Formula

```
verylongint NumHamPath($G, s, t$)
    $V' \leftarrow V \setminus \{s, t\}$
    $N(V') \leftarrow 0$
    foreach $W \subseteq V'$ do
        compute $\bar{N}(W) = P_W[t, n-1]$ using the DP
        $N(V') \leftarrow N(V') + (-1)^{|W|} \bar{N}(W)$
    return $N(V')$
```

$\left\{ \begin{array}{l} O(m^2 n) \\ \text{time} \end{array} \right.$

[Note that numbers in P_W can be exponential in $m!$]

fix!

$$P_W[u,k] = \sum_{\substack{vu \in E \\ v \notin W}} P_W[v, k-1]$$

Runtime: $O(m^2 n \cdot 2^n)$

Space: $O(mn)$ – if we reuse the memory used by the DP in each iteration of the loop.

How can we use this to *find* a Hamiltonian Path?

$\rightsquigarrow$ **Exercise!**

□