Recognizing Sumsets is NP-Complete

Amir Abboud, Nick Fischer, Ron Safier, *Nathan Wallheimer*



INSAIT
Research and Education Foundation

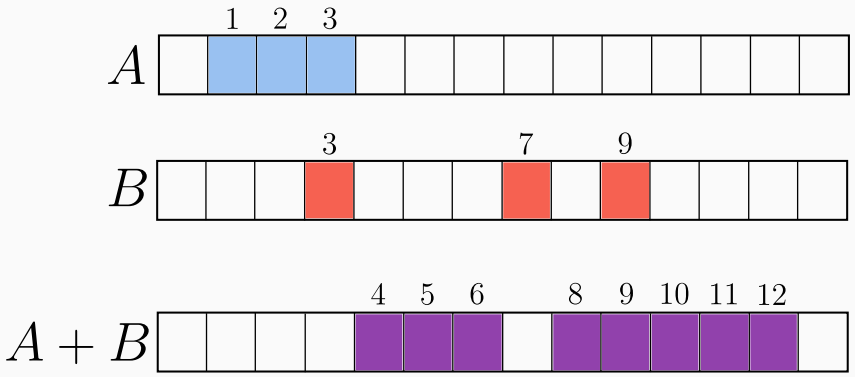
WEIZMANN INSTITUTE OF SCIENCE

Funded by the European Union
European Research Council
Established by the European Commission

Sumsets

Definition (Sum of two sets)

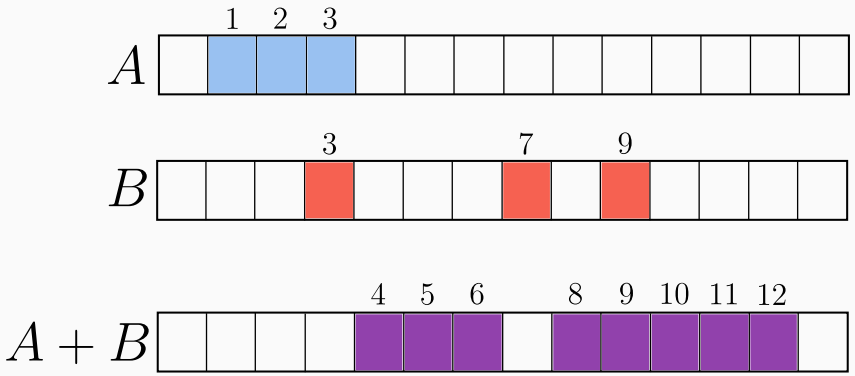
For $A, B \subseteq [0, M] := \{0, 1, \ldots, M\}$, let their sum be $A + B = \{a + b | a \in A, b \in B\}$.



Sumsets

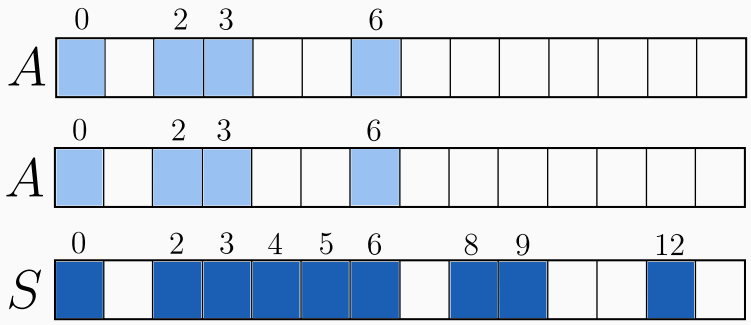
Definition (Sum of two sets)

For $A, B \subseteq [0, M] := \{0, 1, \ldots, M\}$, let their sum be $A + B = \{a + b | a \in A, b \in B\}$.



Definition (Sumset)

A set $S \subseteq [0, M]$ is called a *sumset* if $S = A + A$ for some $A \subseteq [0, M]$.



Sumsets

Motivation: $|A + A|$ is a measure of structure in A.

Sumsets

Motivation: $|A + A|$ is a measure of structure in A.

Example (Arithmetic Progressions)

If $A = [a, b]$, then $A + A = [2a, 2b]$, thus $|A + A| = 2|A| - 1$.
More generally: $|A + A| = 2|A| - 1 \iff A$ is an arithmetic progression.

Sumsets

Motivation: $|A+A|$ is a measure of structure in A.

Example (Arithmetic Progressions)

If $A=[a,b]$, then $A+A=[2a,2b]$, thus $|A+A|=2|A|-1$.
More generally: $|A+A|=2|A|-1 \iff A$ is an arithmetic progression.

Example (Random Sets)

If $A \subseteq [0,M]$ is a small random set, then $|A+A|=\binom{|A|}{2}+|A|$.

Sumsets

Motivation: $|A+A|$ is a measure of structure in A.

Example (Arithmetic Progressions)

If $A=[a,b]$, then $A+A=[2a,2b]$, thus $|A+A|=2|A|-1$.
More generally: $|A+A|=2|A|-1 \iff A$ is an arithmetic progression.

Example (Random Sets)

If $A \subseteq [0,M]$ is a small random set, then $|A+A|=\binom{|A|}{2}+|A|$.

Granville's Question [Croot and Lev, 2007]: Is there an efficient algorithm that given a set S, recognizes if S is a sumset?

Sumsets

Motivation: $|A+A|$ is a measure of structure in A.

Example (Arithmetic Progressions)

If $A=[a,b]$, then $A+A=[2a,2b]$, thus $|A+A|=2|A|-1$.
More generally: $|A+A|=2|A|-1 \iff A$ is an arithmetic progression.

Example (Random Sets)

If $A \subseteq [0,M]$ is a small random set, then $|A+A|=\binom{|A|}{2}+|A|$.

Granville's Question [Croot and Lev, 2007]: Is there an efficient algorithm that given a set S, recognizes if S is a sumset?

Definition (The Sumset Recognition Problem)

Given a set $S \subseteq [0,M]$ of size n, where $M=poly(n)$, decide whether there exists a set $A \subseteq [0,M]$ such that $S=A+A$.

Sumsets

Motivation: $|A+A|$ is a measure of structure in A.

Example (Arithmetic Progressions)

If $A=[a,b]$, then $A+A=[2a,2b]$, thus $|A+A|=2|A|-1$.
More generally: $|A+A|=2|A|-1 \iff A$ is an arithmetic progression.

Example (Random Sets)

If $A \subseteq [0,M]$ is a small random set, then $|A+A|=\binom{|A|}{2}+|A|$.

Granville's Question [Croot and Lev, 2007]: Is there an efficient algorithm that given a set S, recognizes if S is a sumset?

Definition (The Sumset Recognition Problem)

Given a set $S \subseteq [0,M]$ of size n, where $M=poly(n)$, decide whether there exists a set $A \subseteq [0,M]$ such that $S=A+A$.

Motivation:

- Gain a better understanding of the structure of sumsets.

Sumsets

Motivation: $|A+A|$ is a measure of structure in A.

Example (Arithmetic Progressions)

If $A=[a,b]$, then $A+A=[2a,2b]$, thus $|A+A|=2|A|-1$.
More generally: $|A+A|=2|A|-1 \iff A$ is an arithmetic progression.

Example (Random Sets)

If $A \subseteq [0,M]$ is a small random set, then $|A+A|=\binom{|A|}{2}+|A|$.

Granville's Question [Croot and Lev, 2007]: Is there an efficient algorithm that given a set S, recognizes if S is a sumset?

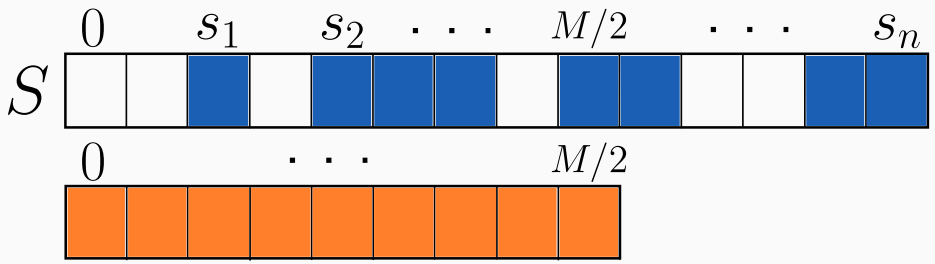
Definition (The Sumset Recognition Problem)

Given a set $S \subseteq [0,M]$ of size n, where $M=poly(n)$, decide whether there exists a set $A \subseteq [0,M]$ such that $S=A+A$.

Motivation:

- Gain a better understanding of the structure of sumsets.
- Gain a better understanding of *factoring* problems.

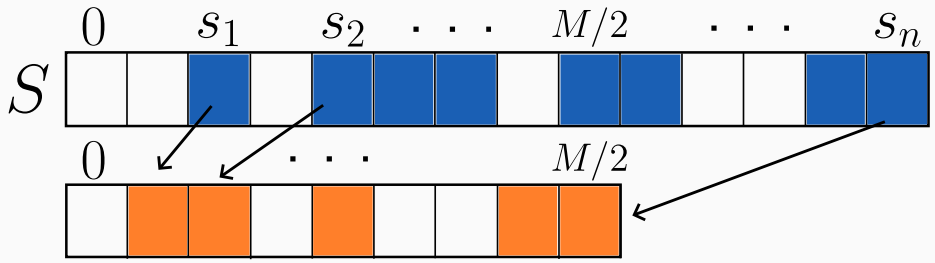
Sumset Recognition Algorithms



An $O^\star(2^{M/2})$-time algorithm

Brute force over all subsets of $[0, M/2]$ and for every $A \subseteq [0, M/2]$, check if $A + A = S$.

Sumset Recognition Algorithms



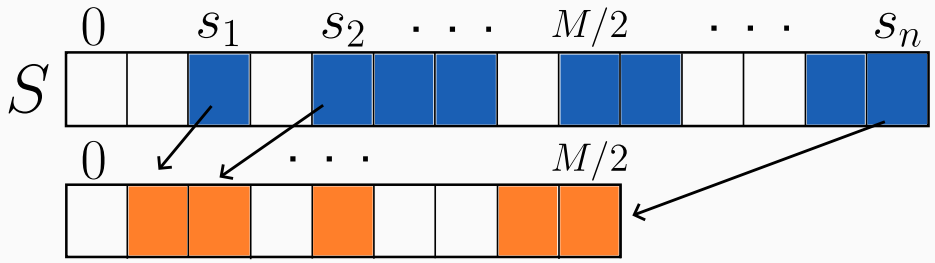
An $O^\star(2^{M/2})$-time algorithm

Brute force over all subsets of $[0, M/2]$ and for every $A \subseteq [0, M/2]$, check if $A + A = S$.

An $O^\star(2^n)$-time algorithm

Brute force only on elements $x \in [0, M/2]$ such that $2x \in S$.

Sumset Recognition Algorithms



An $O^\star(2^{M/2})$-time algorithm

Brute force over all subsets of $[0, M/2]$ and for every $A \subseteq [0, M/2]$, check if $A + A = S$.

An $O^\star(2^n)$-time algorithm

Brute force only on elements $x \in [0, M/2]$ such that $2x \in S$.

Can we do better?

Our Contribution: Recognizing Sumsets is NP-Complete

Theorem (Reduction from 3-SAT)

There is a polynomial-time reduction that, given a 3-SAT formula ϕ with n variables, outputs a set $S \subseteq [0, O(n^4)]$, such that S is a sumset if and only if ϕ is satisfiable.

Our Contribution: Recognizing Sumsets is NP-Complete

Theorem (Reduction from 3-SAT)

There is a polynomial-time reduction that, given a 3-SAT formula ϕ with n variables, outputs a set $S \subseteq [0, O(n^4)]$, such that S is a sumset if and only if ϕ is satisfiable.

Corollary (Exponential Time Hypothesis)

Assuming that solving 3-SAT requires $2^{\Omega(n)}$ time, then solving Sumset Recognition requires $2^{\Omega(M^{1/4})}$ time, and therefore also $2^{\Omega(n^{1/4})}$ time.

Our Contribution: Recognizing Sumsets is NP-Complete

Theorem (Reduction from 3-SAT)

There is a polynomial-time reduction that, given a 3-SAT formula ϕ with n variables, outputs a set $S \subseteq [0, O(n^4)]$, such that S is a sumset if and only if ϕ is satisfiable.

Corollary (Exponential Time Hypothesis)

Assuming that solving 3-SAT requires $2^{\Omega(n)}$ time, then solving Sumset Recognition requires $2^{\Omega(M^{1/4})}$ time, and therefore also $2^{\Omega(n^{1/4})}$ time.

Open Question: Resolve the gap between $2^{O(n)}$ and $2^{\Omega(n^{1/4})}$.

Proof Outline

We are given a 3-SAT formula ϕ with n variables and m clauses.
Goal: Design a set $S := S(\phi)$, and sets $\{A_\alpha\}_{\alpha \in \{0,1\}^n}$, such that:

$$\phi \text{ is satisfiable} \quad \Longleftrightarrow \quad S \text{ is a sumset.}$$

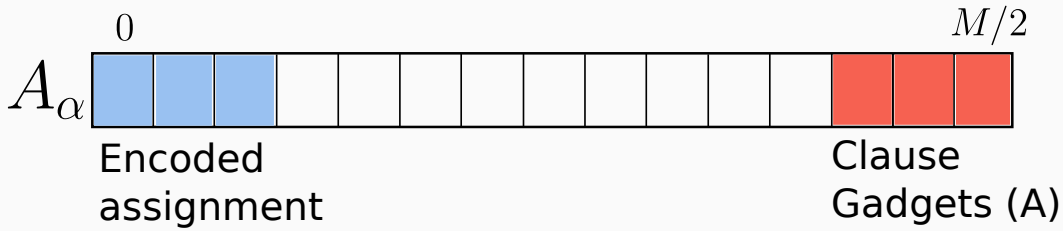
Moreover, $S = A_\alpha + A_\alpha$ for any satisfying assignment $\alpha \in \{0,1\}^n$.

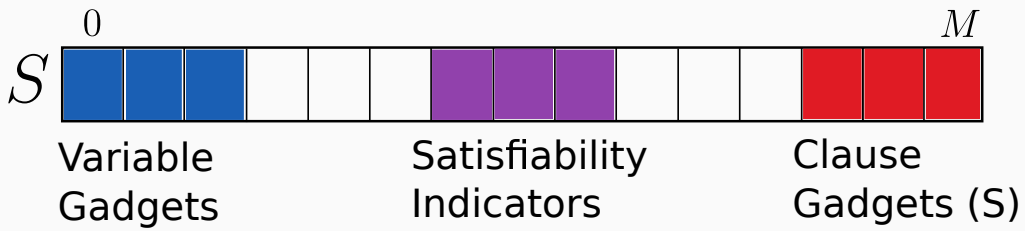
Proof Outline

We are given a 3-SAT formula ϕ with n variables and m clauses.
Goal: Design a set $S := S(\phi)$, and sets $\{A_\alpha\}_{\alpha \in \{0,1\}^n}$, such that:

$$\phi \text{ is satisfiable} \iff S \text{ is a sumset.}$$

Moreover, $S = A_\alpha + A_\alpha$ for any satisfying assignment $\alpha \in \{0,1\}^n$.



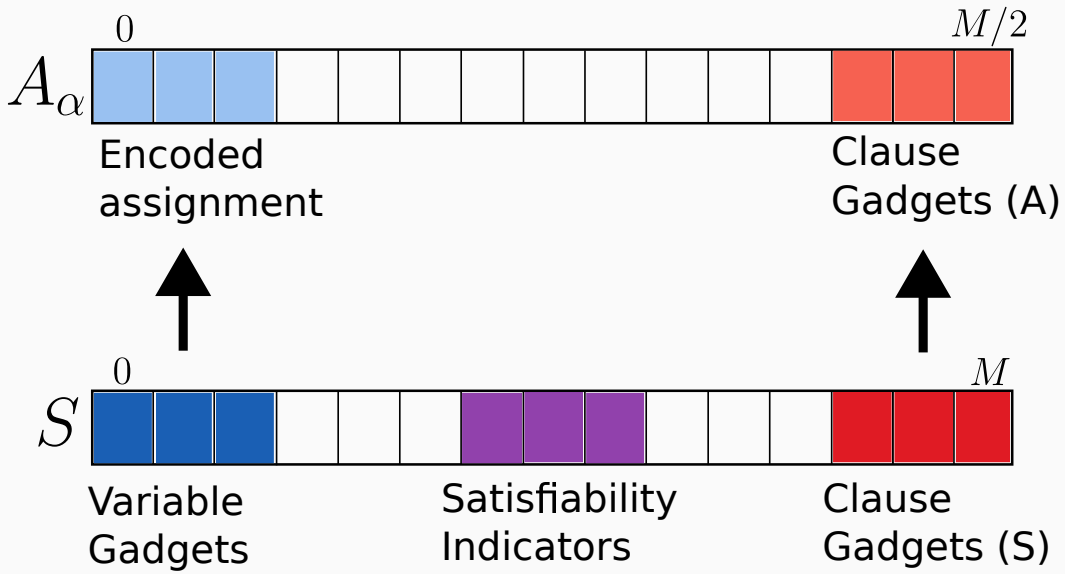


Proof Outline

We are given a 3-SAT formula ϕ with n variables and m clauses.
Goal: Design a set $S := S(\phi)$, and sets $\{A_\alpha\}_{\alpha \in \{0,1\}^n}$, such that:

$$\phi \text{ is satisfiable} \iff S \text{ is a sumset.}$$

Moreover, $S = A_\alpha + A_\alpha$ for any satisfying assignment $\alpha \in \{0,1\}^n$.

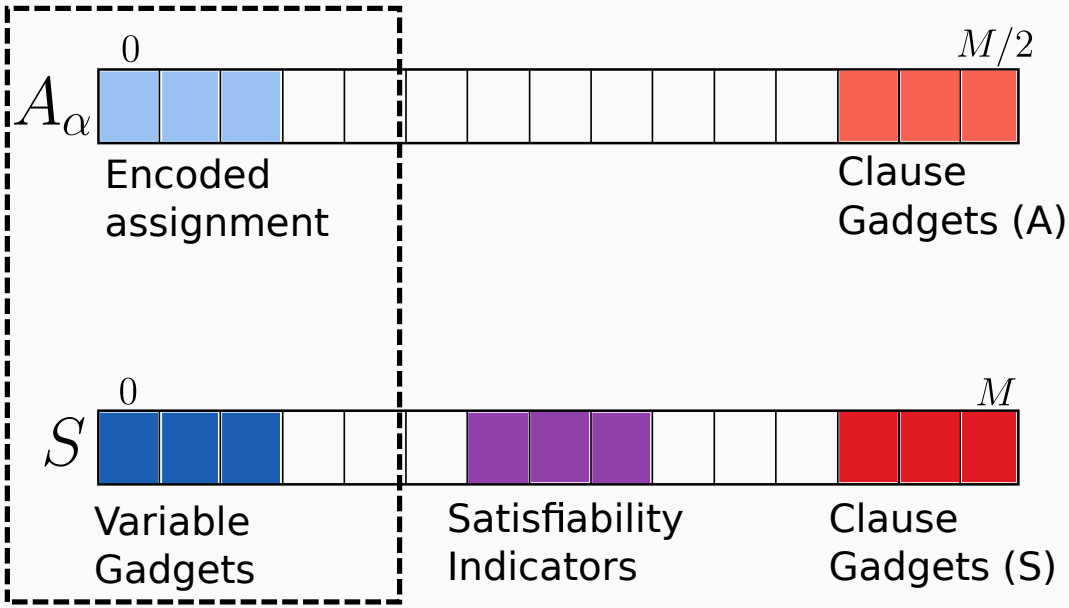


Proof Outline

We are given a 3-SAT formula ϕ with n variables and m clauses.
Goal: Design a set $S := S(\phi)$, and sets $\{A_\alpha\}_{\alpha \in \{0,1\}^n}$, such that:

$$\phi \text{ is satisfiable} \iff S \text{ is a sumset.}$$

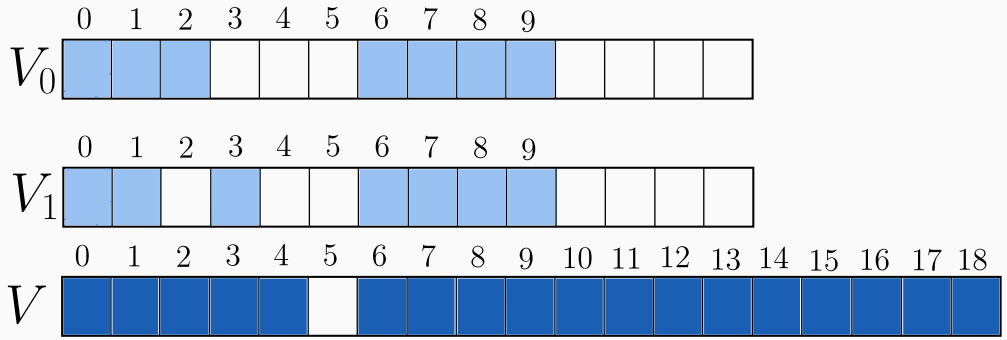
Moreover, $S = A_\alpha + A_\alpha$ for any satisfying assignment $\alpha \in \{0,1\}^n$.



Variable Gadgets

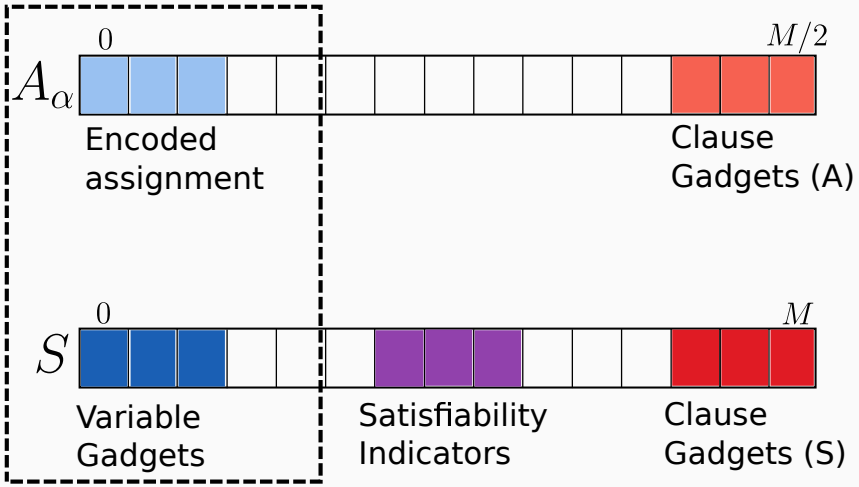
Goal: Design a constant sized set V that has only two representations: $V = V_0 + V_0$ and $V = V_1 + V_1$.

$$V_0 = \{0, 1, 2, 6, 7, 8, 9\}$$
$$V_1 = \{0, 1, 3, 6, 7, 8, 9\}$$
$$V = [0, 18] \setminus \{5\}$$

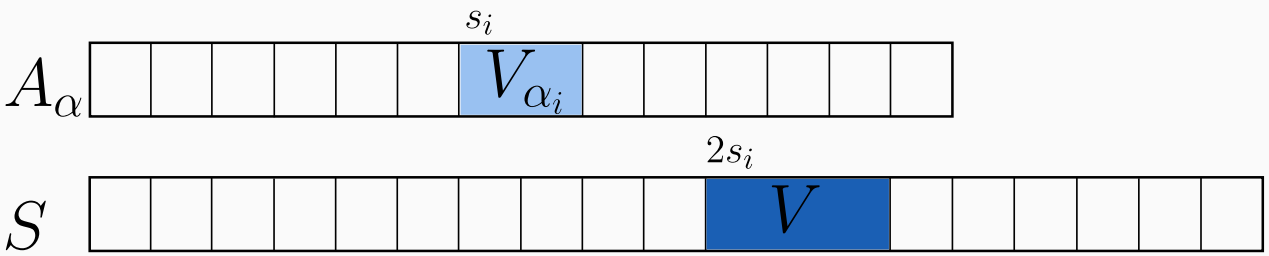


Remark: This is the smallest variable gadget.

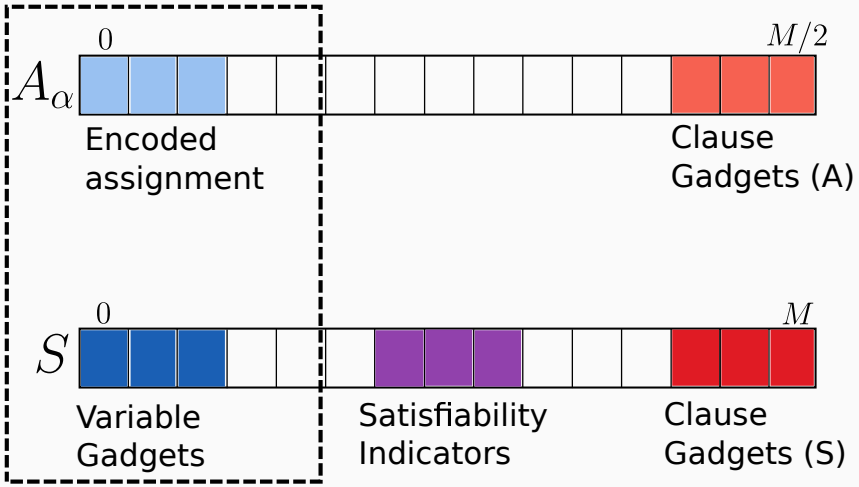
Variable Gadgets



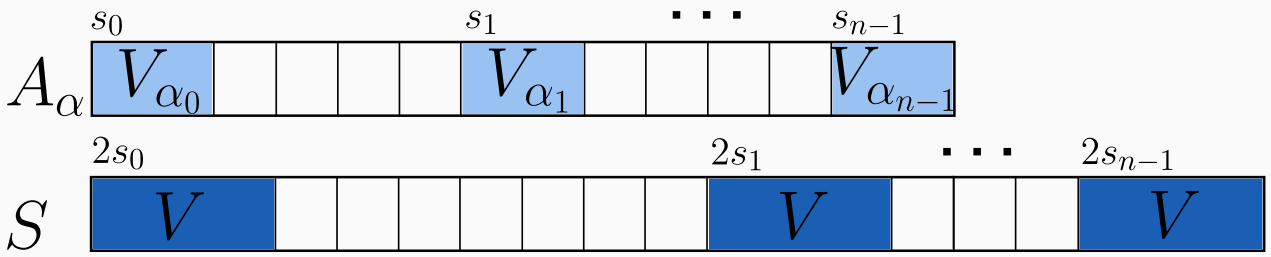
- Encoding one variable assignment at position s_i:



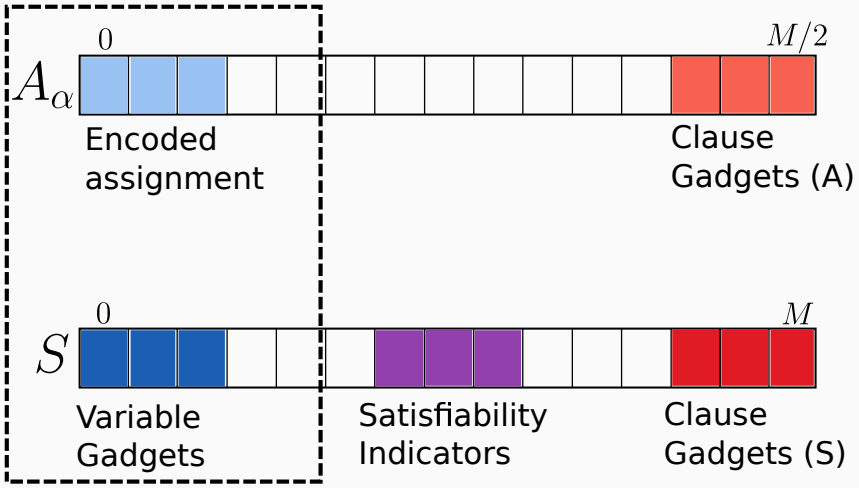
Variable Gadgets



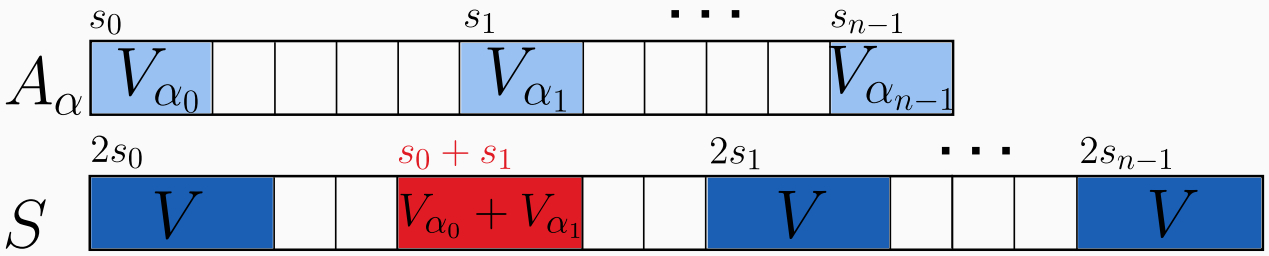
- Encoding one variable assignment at position s_i:
- Encoding n variable assignments at positions $s_0 < \cdots < s_{n-1}$:



Variable Gadgets

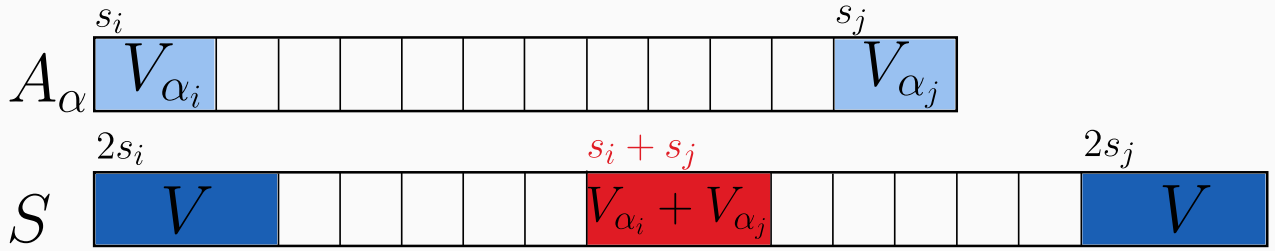


- Encoding one variable assignment at position s_i:
- Encoding n variable assignments at positions $s_0 < \cdots < s_{n-1}$:



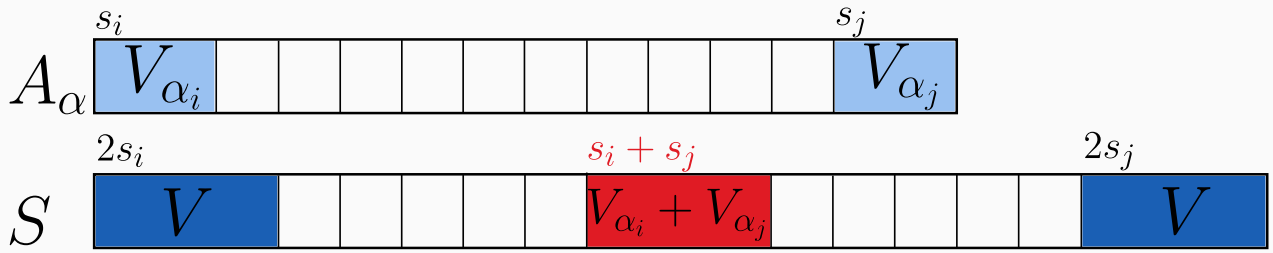
The cross-term $V_{\alpha_0} + V_{\alpha_1}$ is an unwanted by-product.

Cross-Terms



The cross-term $V_{\alpha_i} + V_{\alpha_j}$ appears at $s_i + s_j$ for every $i < j$. There are two problems with it:

Cross-Terms



The cross-term $V_{\alpha_i} + V_{\alpha_j}$ appears at $s_i + s_j$ for every $i < j$. There are two problems with it:

1. $s_i + s_j$ may collide with legitimate positions, i.e. $2s_k$. This problem can be solved by picking random $s_0, \ldots, s_{n-1}$, or deterministically, using *explicit* Sidon sets.

Explicit Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if all pairwise sums $s_i + s_j$, for $i \leq j$, are distinct.

Explicit Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if all pairwise sums $s_i + s_j$, for $i \leq j$, are distinct.

Greedy construction (Mian-Chowla sequence): The greedy sequence $0, 1, 3, 7, \ldots$ gives a Sidon set in $poly(n)$ time, with $s_{n-1} = O(n^3)$.

Explicit Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if all pairwise sums $s_i + s_j$, for $i \leq j$, are distinct.

Greedy construction (Mian-Chowla sequence): The greedy sequence $0, 1, 3, 7, \ldots$ gives a Sidon set in $poly(n)$ time, with $s_{n-1} = O(n^3)$.

Lemma (Erdős and Turán.)

In time $poly(n)$, we can compute a Sidon set of size n over $[0, O(n^2)]$.

Explicit Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if all pairwise sums $s_i + s_j$, for $i \leq j$, are distinct.

Greedy construction (Mian-Chowla sequence): The greedy sequence $0, 1, 3, 7, \ldots$ gives a Sidon set in $poly(n)$ time, with $s_{n-1} = O(n^3)$.

Lemma (Erdős and Turán.)

In time $poly(n)$, we can compute a Sidon set of size n over $[0, O(n^2)]$.

1. For a prime p, the set $\{(x, x^2) | x \in \mathbb{F}_p\}$ is a Sidon set over $\mathbb{F}_p^2$.

Explicit Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if all pairwise sums $s_i + s_j$, for $i \leq j$, are distinct.

Greedy construction (Mian-Chowla sequence): The greedy sequence $0, 1, 3, 7, \ldots$ gives a Sidon set in $poly(n)$ time, with $s_{n-1} = O(n^3)$.

Lemma (Erdős and Turán.)

In time $poly(n)$, we can compute a Sidon set of size n over $[0, O(n^2)]$.

1. For a prime p, the set $\{(x, x^2) | x \in \mathbb{F}_p\}$ is a Sidon set over $\mathbb{F}_p^2$.

$$(x, x^2) + (y, y^2) = (z, z^2) + (w, w^2) \iff \{x, y\} = \{z, w\}.$$

Explicit Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if all pairwise sums $s_i + s_j$, for $i \leq j$, are distinct.

Greedy construction (Mian-Chowla sequence): The greedy sequence $0, 1, 3, 7, \ldots$ gives a Sidon set in $poly(n)$ time, with $s_{n-1} = O(n^3)$.

Lemma (Erdős and Turán.)

In time $poly(n)$, we can compute a Sidon set of size n over $[0, O(n^2)]$.

1. For a prime p, the set $\{(x, x^2) | x \in \mathbb{F}_p\}$ is a Sidon set over $\mathbb{F}_p^2$.

$$(x, x^2) + (y, y^2) = (z, z^2) + (w, w^2) \iff \{x, y\} = \{z, w\}.$$

2. Embed $\mathbb{F}_p^2$ into the integers using the map $(x, y) \mapsto 2p \cdot x + y$.

Explicit Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if all pairwise sums $s_i + s_j$, for $i \leq j$, are distinct.

Greedy construction (Mian-Chowla sequence): The greedy sequence $0, 1, 3, 7, \ldots$ gives a Sidon set in $poly(n)$ time, with $s_{n-1} = O(n^3)$.

Lemma (Erdős and Turán.)

In time $poly(n)$, we can compute a Sidon set of size n over $[0, O(n^2)]$.

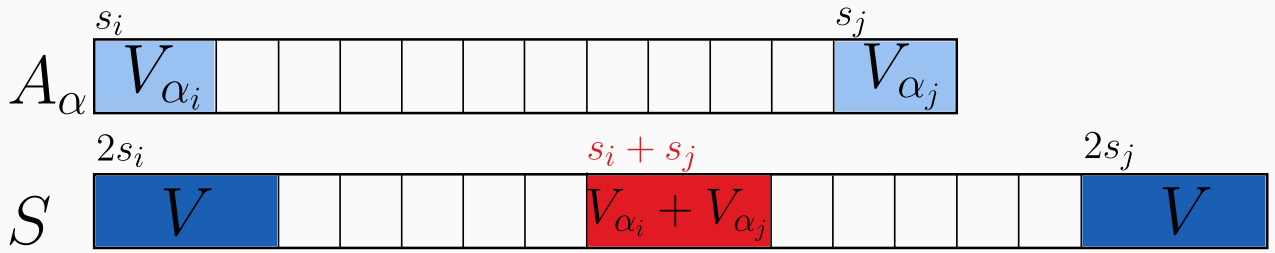
1. For a prime p, the set $\{(x, x^2) | x \in \mathbb{F}_p\}$ is a Sidon set over $\mathbb{F}_p^2$.

$$(x, x^2) + (y, y^2) = (z, z^2) + (w, w^2) \iff \{x, y\} = \{z, w\}.$$

2. Embed $\mathbb{F}_p^2$ into the integers using the map $(x, y) \mapsto 2p \cdot x + y$.

We get a Sidon set of size p over $[0, O(p^2)]$.

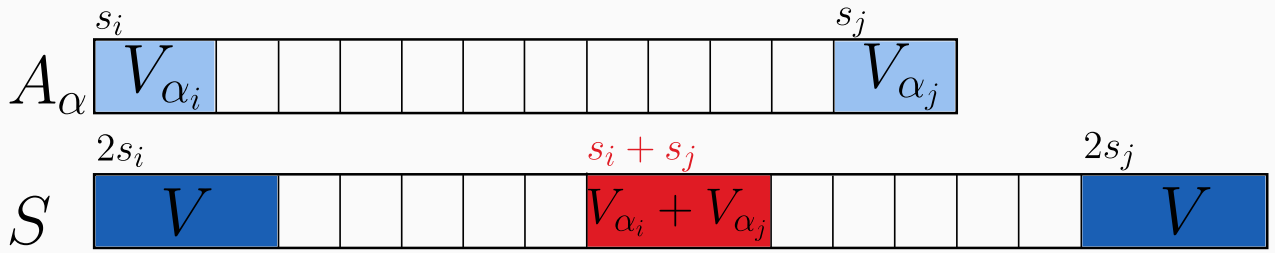
Cross-Terms



The cross-term $V_{\alpha_i} + V_{\alpha_j}$ appears at $s_i + s_j$ for every $i < j$. There are two problems with it:

1. $s_i + s_j$ may collide with legitimate positions, i.e. $2s_k$. This problem can be solved by picking random $s_0, \ldots, s_{n-1}$, or deterministically, using *explicit* Sidon sets.

Cross-Terms

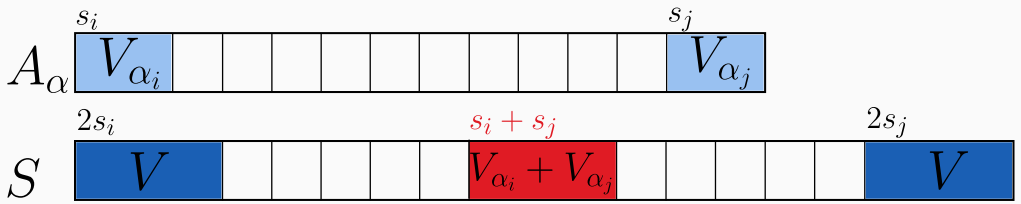


The cross-term $V_{\alpha_i} + V_{\alpha_j}$ appears at $s_i + s_j$ for every $i < j$. There are two problems with it:

1. $s_i + s_j$ may collide with legitimate positions, i.e. $2s_k$. This problem can be solved by picking random $s_0, \ldots, s_{n-1}$, or deterministically, using *explicit* Sidon sets.

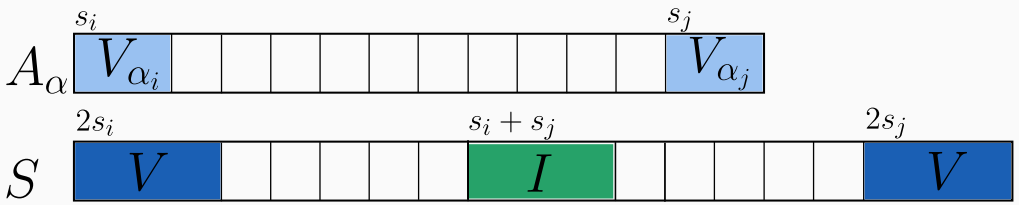
1. Having $V_{\alpha_i} + V_{\alpha_j}$ in S will make S dependent on α. The set S has to be *oblivious* to α.

Masking



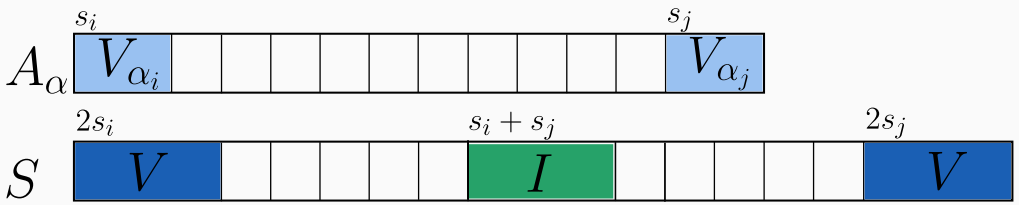
Solution: Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

Masking



Solution: Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

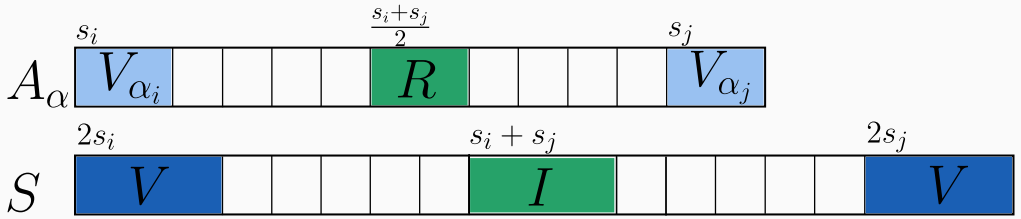
Masking



Solution: Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

- Since $V_{\alpha_i} + V_{\alpha_j} \subset I$, R has to appear in A_α.

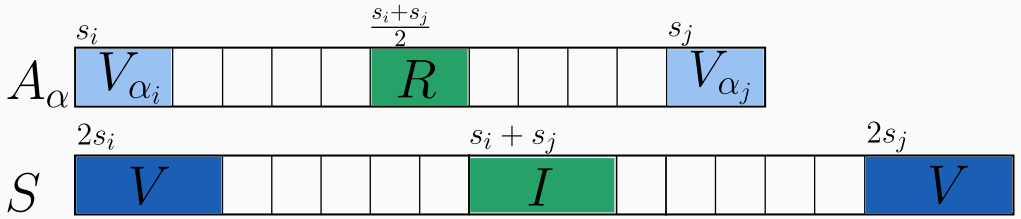
Masking



Solution: Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

- Since $V_{\alpha_i} + V_{\alpha_j} \subset I$, R has to appear in A_α.

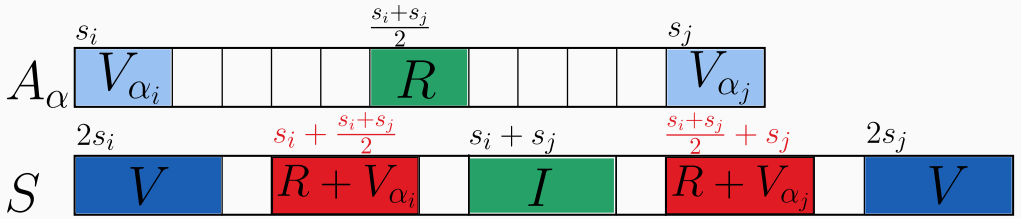
Masking



Solution: Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

- Since $V_{\alpha_i} + V_{\alpha_j} \subset I$, R has to appear in A_α.
- New cross-terms: $R + V_{\alpha_i}$ and $R + V_{\alpha_j}$.

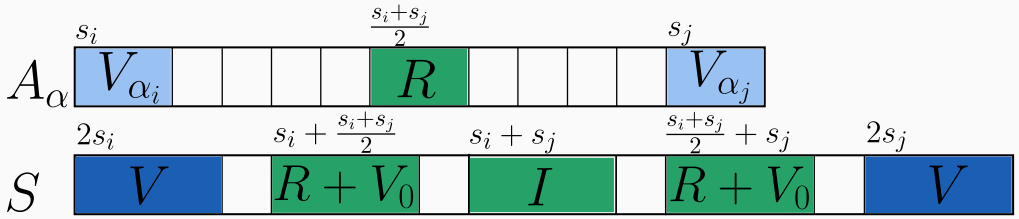
Masking



Solution: Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

- Since $V_{\alpha_i} + V_{\alpha_j} \subset I$, R has to appear in A_α.
- New cross-terms: $R + V_{\alpha_i}$ and $R + V_{\alpha_j}$.

Masking

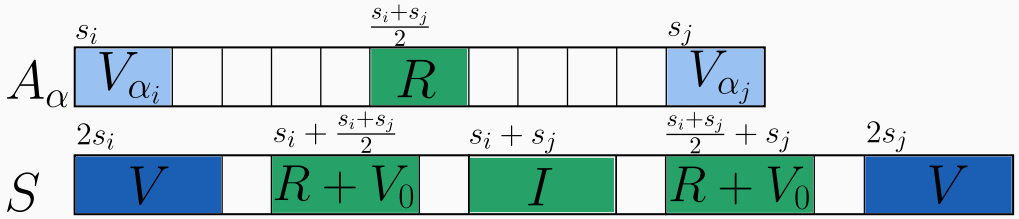


Solution: Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

- Since $V_{\alpha_i} + V_{\alpha_j} \subset I$, R has to appear in A_α.
- New cross-terms: $R + V_{\alpha_i}$ and $R + V_{\alpha_j}$.

Key idea: $R + V_{\alpha_j}$ is an interval that's independent of α, so we can place $R + V_0$.

Masking



Solution: Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

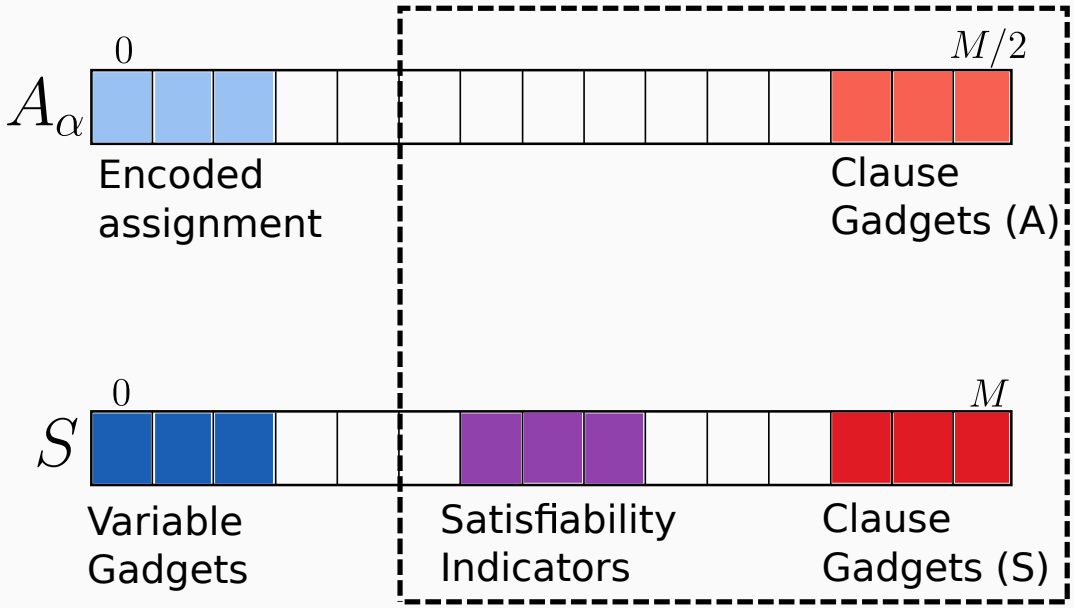
- Since $V_{\alpha_i} + V_{\alpha_j} \subset I$, R has to appear in A_α.
- New cross-terms: $R + V_{\alpha_i}$ and $R + V_{\alpha_j}$.

Key idea: $R + V_{\alpha_j}$ is an interval that's independent of α, so we can place $R + V_0$.

Masking (informally)

Whenever our design of S and A_α is such that $A_\alpha + A_\alpha$ also contains elements not oblivious to α, we can employ masking to resolve it.

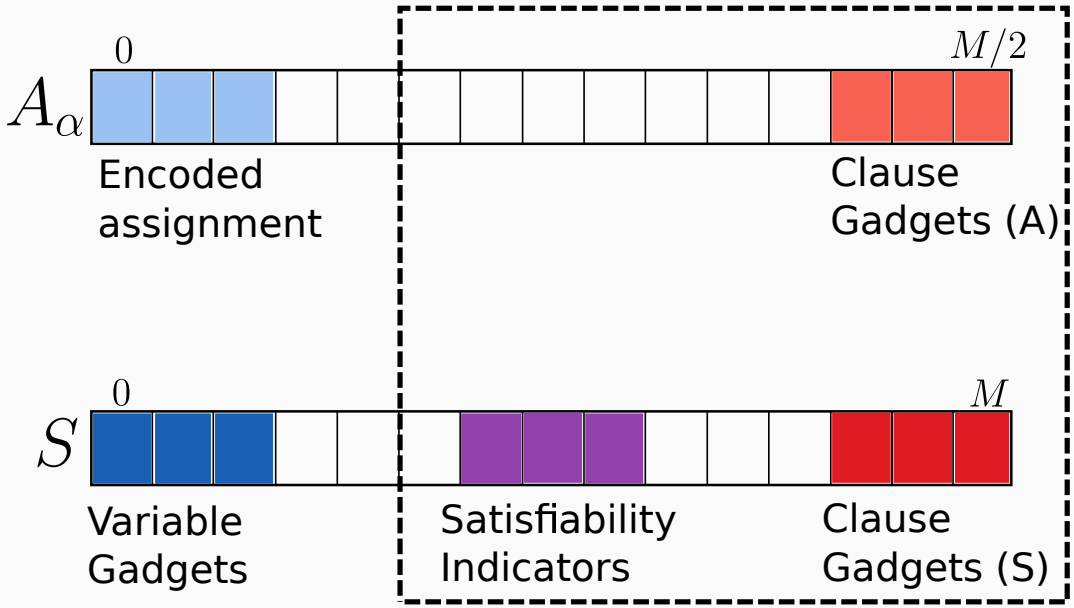
Clause Gadgets



Goal: For every clause C_k, have a gadget $G_k \subseteq A_\alpha$, and a number $t_k \in S$, so that:

$$C_k \text{ is satisfied by } \alpha \iff <\alpha> + G_k \ni t_k.$$

Clause Gadgets



Goal: For every clause C_k, have a gadget $G_k \subseteq A_\alpha$, and a number $t_k \in S$, so that:

$$C_k \text{ is satisfied by } \alpha \iff < \alpha > + G_k \ni t_k.$$

Remark: To enforce G_k into A_α, let S contain $G_k + G_k$.

Clause Gadgets

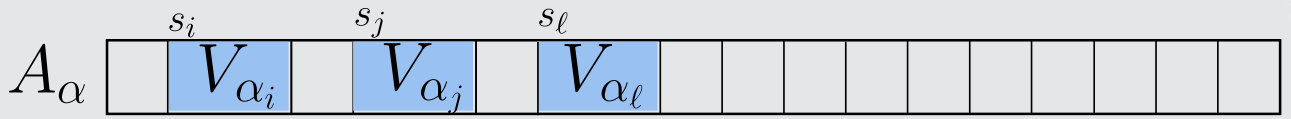
Example (The gadget G_k)

Let $C_k = (x_i \vee \bar{x}_j \vee x_\ell)$.

Clause Gadgets

Example (The gadget G_k)

Let $C_k = (x_i \vee \bar{x}_j \vee x_\ell)$.

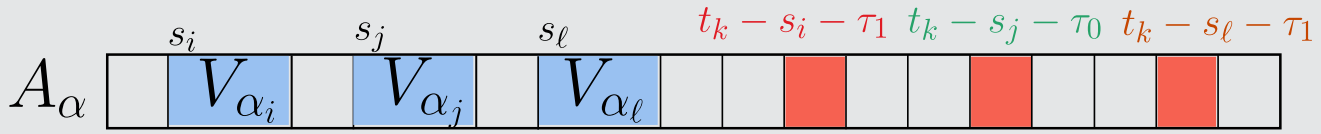


Recall: There are unique $\tau_0 \in V_0 \setminus V_1$ and $\tau_1 \in V_1 \setminus V_0$.

Clause Gadgets

Example (The gadget G_k)

Let $C_k = (x_i \vee \bar{x}_j \vee x_\ell)$.



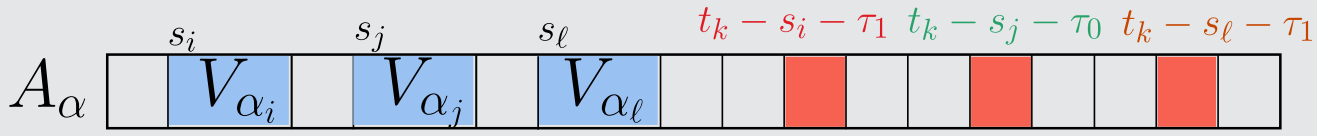
Recall: There are unique $\tau_0 \in V_0 \setminus V_1$ and $\tau_1 \in V_1 \setminus V_0$.
Let $t_k \gg s_{n-1}$ and:

$$G_k = \{t_k - s_i - \tau_1, t_k - s_j - \tau_0, t_k - s_\ell - \tau_1\}$$

Clause Gadgets

Example (The gadget G_k)

Let $C_k = (x_i \vee \bar{x}_j \vee x_\ell)$.



Recall: There are unique $\tau_0 \in V_0 \setminus V_1$ and $\tau_1 \in V_1 \setminus V_0$.
Let $t_k \gg s_{n-1}$ and:

$$G_k = \{t_k - s_i - \tau_1, t_k - s_j - \tau_0, t_k - s_\ell - \tau_1\}$$

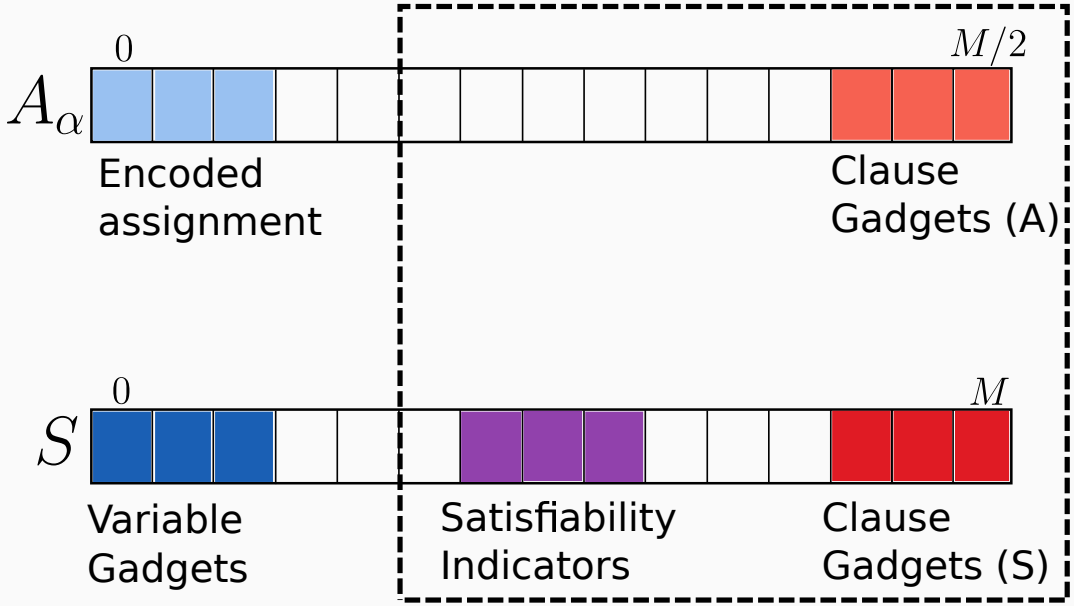
Observation

$A_\alpha + A_\alpha$ contains the set $(s_i + V_{\alpha_i}) + (t_k - s_i - \tau_1)$.

- s_i and $-s_i$ cancel out.
- $-\tau_1$ cancels if and only if $V_{\alpha_i} = V_1$, i.e., if and only if α_i satisfies the clause.

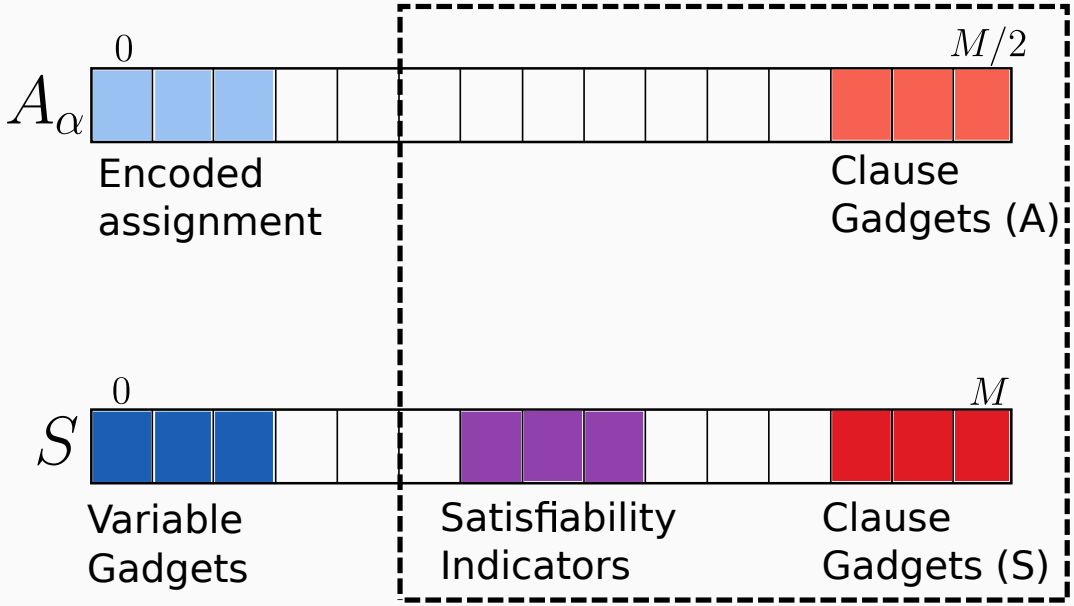
Hence, t_k is in the set if and only if α satisfies the clause.

Clause Gadgets



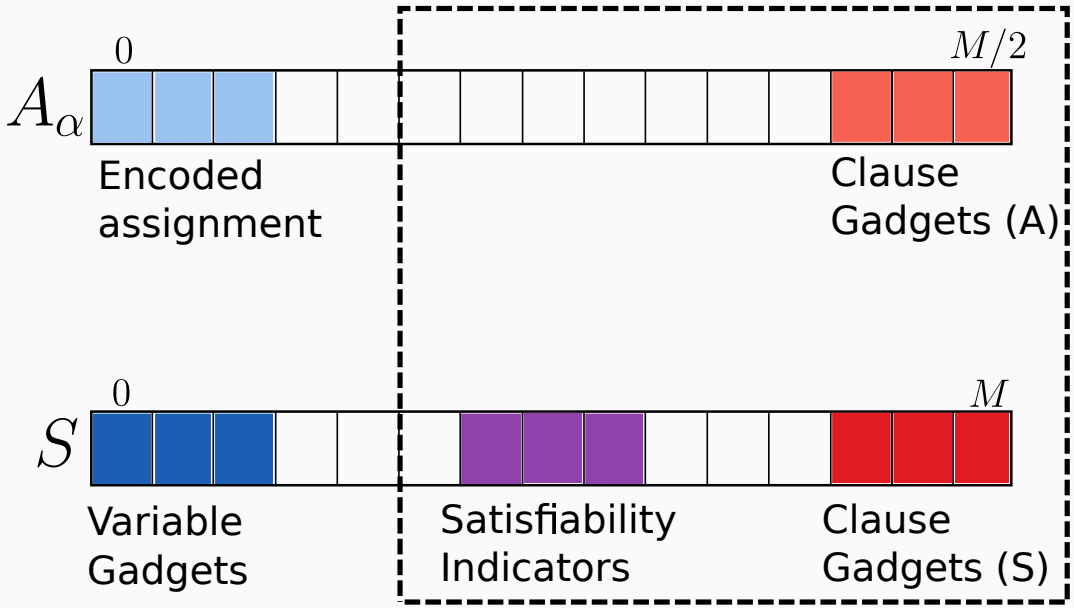
- Add $t_0 < t_1 < \ldots < t_{m-1}$ to S.

Clause Gadgets



- Add $t_0 < t_1 < \ldots < t_{m-1}$ to S.
- For every $k \in [m]$, have G_k in A_α.

Clause Gadgets



- Add $t_0 < t_1 < \ldots < t_{m-1}$ to S.
- For every $k \in [m]$, have G_k in A_α.

Problem: How could S enforce G_k into A_α?

Positioning

Positioning: How could S enforce some set G into A_α?

Positioning

Positioning: How could S enforce some set G into A_α?
Having $G + G \subseteq S$ is not good enough, since it could be that $G + G = R + R$ for some $R \neq G$.

Positioning

Positioning: How could S enforce some set G into A_α?
Having $G + G \subseteq S$ is not good enough, since it could be that $G + G = R + R$ for some $R \neq G$.

Sumsets with a unique representation

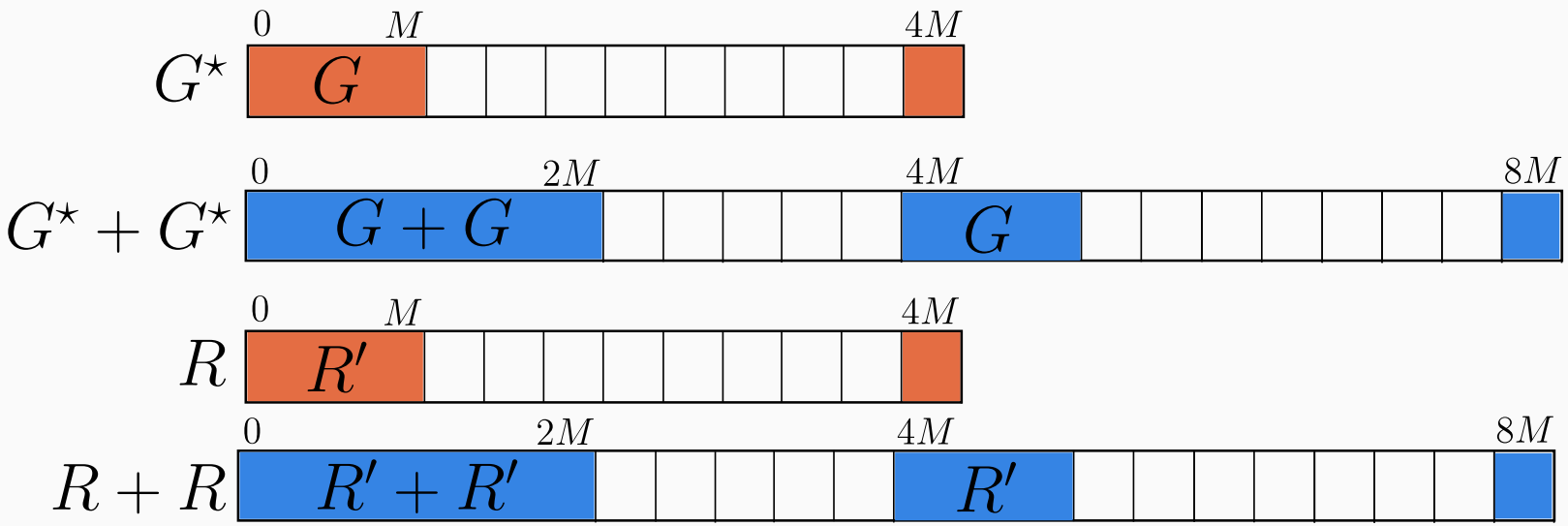
For any set $G \subseteq [0, M]$, the set $G^\star := G \cup \{4M\}$ is such that $G^\star + G^\star = R + R \iff R = G^\star$.

Positioning

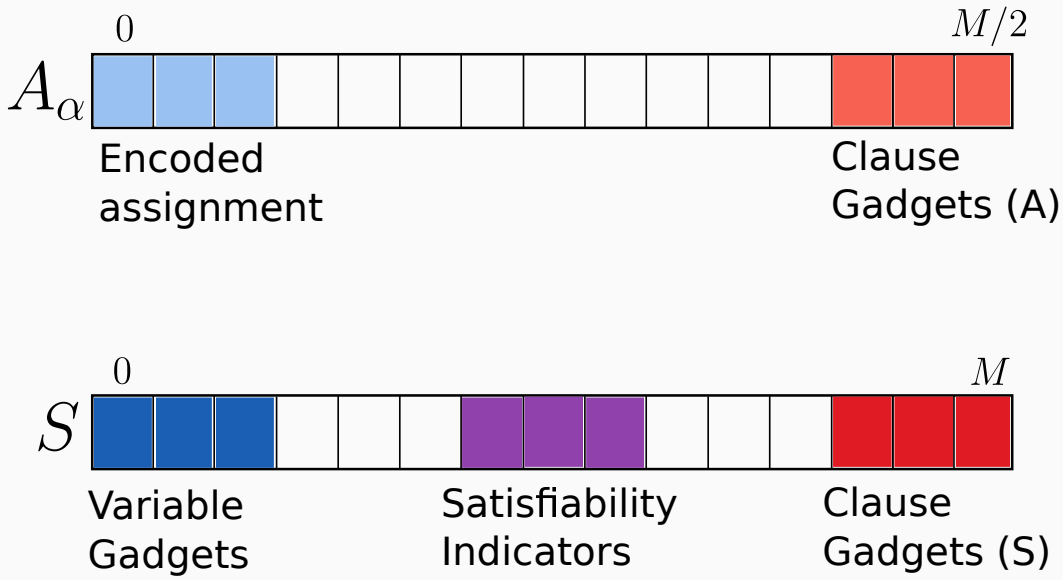
Positioning: How could S enforce some set G into A_α?
Having $G + G \subseteq S$ is not good enough, since it could be that $G + G = R + R$ for some $R \neq G$.

Sumsets with a unique representation

For any set $G \subseteq [0, M]$, the set $G^\star := G \cup \{4M\}$ is such that $G^\star + G^\star = R + R \iff R = G^\star$.



Masking and Positioning, Generalized



Our actual construction is more modular and uses generalized masking and positioning lemmas:

- Masking: Can ignore by-products that depend on α in $A_\alpha + A_\alpha$.
- Positioning: Can ignore solutions $S = A + A$ for $A \notin \{A_\alpha\}_{\alpha \in \{0,1\}^n}$.

Open Question 1: Closing the Gap

Closing the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?

Open Question 1: Closing the Gap

Closing the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?

- Current best upper bound: An $O^\star(3^{n/3})$ algorithm.

Open Question 1: Closing the Gap

Closing the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?

- Current best upper bound: An $O^\star(3^{n/3})$ algorithm.
- Lower bound: The $O(n^4)$ blowup is due to an $O(n^2)$ blowup from Sidon sets, and an additional $O(n^2)$ blowup from masking.

Open Question 1: Closing the Gap

Closing the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?

- Current best upper bound: An $O^\star(3^{n/3})$ algorithm.
- Lower bound: The $O(n^4)$ blowup is due to an $O(n^2)$ blowup from Sidon sets, and an additional $O(n^2)$ blowup from masking.

Sumsets avoiding the legitimate positions

Goal: Compute $S = \{s_0, \ldots, s_{n-1}\}$ and $T = \{t_0, \ldots, t_{m-1}\}$, such that:

Open Question 1: Closing the Gap

Closing the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?

- Current best upper bound: An $O^\star(3^{n/3})$ algorithm.
- Lower bound: The $O(n^4)$ blowup is due to an $O(n^2)$ blowup from Sidon sets, and an additional $O(n^2)$ blowup from masking.

Sumsets avoiding the legitimate positions

Goal: Compute $S = \{s_0, \ldots, s_{n-1}\}$ and $T = \{t_0, \ldots, t_{m-1}\}$, such that:

1. No solutions to $s_i + s_j = 2s_k$.

Open Question 1: Closing the Gap

Closing the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?

- Current best upper bound: An $O^\star(3^{n/3})$ algorithm.
- Lower bound: The $O(n^4)$ blowup is due to an $O(n^2)$ blowup from Sidon sets, and an additional $O(n^2)$ blowup from masking.

Sumsets avoiding the legitimate positions

Goal: Compute $S = \{s_0, \ldots, s_{n-1}\}$ and $T = \{t_0, \ldots, t_{m-1}\}$, such that:

1. No solutions to $s_i + s_j = 2s_k$.
2. No solutions to $t_k = (t_\ell - s_j) + s_i$, where C_ℓ contains the variable x_j.

Open Question 1: Closing the Gap

Closing the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?

- Current best upper bound: An $O^\star(3^{n/3})$ algorithm.
- Lower bound: The $O(n^4)$ blowup is due to an $O(n^2)$ blowup from Sidon sets, and an additional $O(n^2)$ blowup from masking.

Sumsets avoiding the legitimate positions

Goal: Compute $S = \{s_0, \ldots, s_{n-1}\}$ and $T = \{t_0, \ldots, t_{m-1}\}$, such that:

1. No solutions to $s_i + s_j = 2s_k$.
2. No solutions to $t_k = (t_\ell - s_j) + s_i$, where C_ℓ contains the variable x_j.

- *Behrend's construction*: $\{s_0, \ldots, s_{n-1}\} \subseteq [0, n^{1+o(1)}]$ with no solutions to $s_i + s_j = 2s_k$, except for $i = j = k$.

Open Question 1: Closing the Gap

Closing the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?

- Current best upper bound: An $O^\star(3^{n/3})$ algorithm.
- Lower bound: The $O(n^4)$ blowup is due to an $O(n^2)$ blowup from Sidon sets, and an additional $O(n^2)$ blowup from masking.

Sumsets avoiding the legitimate positions

Goal: Compute $S = \{s_0, \ldots, s_{n-1}\}$ and $T = \{t_0, \ldots, t_{m-1}\}$, such that:

1. No solutions to $s_i + s_j = 2s_k$.
2. No solutions to $t_k = (t_\ell - s_j) + s_i$, where C_ℓ contains the variable x_j.

- *Behrend's construction*: $\{s_0, \ldots, s_{n-1}\} \subseteq [0, n^{1+o(1)}]$ with no solutions to $s_i + s_j = 2s_k$, except for $i = j = k$.
- What about the second condition?

Graphical Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if it contains no solutions to $s_i + s_j = s_k + s_\ell$, except for $\{i, j\} = \{k, \ell\}$.

Definition ($3AP$-free set)

A set $\{s_0, \ldots, s_{n-1}\}$ is $3AP$-free if it contains no solutions to $s_i + s_j = 2s_k$, except for $i = j = k$.

Graphical Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if it contains no solutions to $s_i + s_j = s_k + s_\ell$, except for $\{i, j\} = \{k, \ell\}$.

Definition ($3AP$-free set)

A set $\{s_0, \ldots, s_{n-1}\}$ is $3AP$-free if it contains no solutions to $s_i + s_j = 2s_k$, except for $i = j = k$.

Definition (Graphical Sidon set)

Given a graph $G = ([n], E)$, a set $\{s_0, \ldots, s_{n-1}\}$ is a G-Sidon set if it contains no solutions to $s_u + s_v = s_i + s_j$, for every $uv \in E$ and $\{i, j\} \neq \{u, v\}$.

Graphical Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if it contains no solutions to $s_i + s_j = s_k + s_\ell$, except for $\{i, j\} = \{k, \ell\}$.

Definition ($3AP$-free set)

A set $\{s_0, \ldots, s_{n-1}\}$ is $3AP$-free if it contains no solutions to $s_i + s_j = 2s_k$, except for $i = j = k$.

Definition (Graphical Sidon set)

Given a graph $G = ([n], E)$, a set $\{s_0, \ldots, s_{n-1}\}$ is a G-Sidon set if it contains no solutions to $s_u + s_v = s_i + s_j$, for every $uv \in E$ and $\{i, j\} \neq \{u, v\}$.

- $3AP$-free set: $G = n$ self-loops.

Graphical Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if it contains no solutions to $s_i + s_j = s_k + s_\ell$, except for $\{i, j\} = \{k, \ell\}$.

Definition ($3AP$-free set)

A set $\{s_0, \ldots, s_{n-1}\}$ is $3AP$-free if it contains no solutions to $s_i + s_j = 2s_k$, except for $i = j = k$.

Definition (Graphical Sidon set)

Given a graph $G = ([n], E)$, a set $\{s_0, \ldots, s_{n-1}\}$ is a G-Sidon set if it contains no solutions to $s_u + s_v = s_i + s_j$, for every $uv \in E$ and $\{i, j\} \neq \{u, v\}$.

- $3AP$-free set: $G = n$ self-loops.
- Sidon set: $G =$ clique.

Graphical Sidon Sets

Definition (Sidon set)

A set $\{s_0, \cdots, s_{n-1}\}$ is called a Sidon set if it contains no solutions to $s_i + s_j = s_k + s_\ell$, except for $\{i, j\} = \{k, \ell\}$.

Definition ($3AP$-free set)

A set $\{s_0, \ldots, s_{n-1}\}$ is $3AP$-free if it contains no solutions to $s_i + s_j = 2s_k$, except for $i = j = k$.

Definition (Graphical Sidon set)

Given a graph $G = ([n], E)$, a set $\{s_0, \ldots, s_{n-1}\}$ is a G-Sidon set if it contains no solutions to $s_u + s_v = s_i + s_j$, for every $uv \in E$ and $\{i, j\} \neq \{u, v\}$.

- $3AP$-free set: $G = n$ self-loops.
- Sidon set: $G =$ clique.

What if G is bounded degree? Can we still get $S \subseteq [0, n^{1+o(1)}]$?

Open Question 2: Generalizations

- Is it NP-Complete to decide if there exists A, B, $|A|, |B| > 1$, such that $A + B = S$?

Open Question 2: Generalizations

- Is it NP-Complete to decide if there exists A, B, $|A|, |B| > 1$, such that $A + B = S$?
- Is it NP-Complete to decide if exists A such that $A + A + A = S$?

Open Question 2: Generalizations

- Is it NP-Complete to decide if there exists A, B, $|A|, |B| > 1$, such that $A + B = S$?
- Is it NP-Complete to decide if exists A such that $A + A + A = S$?
- Approximate version? e.g. can we obtain a good approximation to $\min_A |S\Delta(A + A)|$?

Open Question 2: Generalizations

- Is it NP-Complete to decide if there exists A, B, $|A|, |B| > 1$, such that $A + B = S$?
- Is it NP-Complete to decide if exists A such that $A + A + A = S$?
- Approximate version? e.g. can we obtain a good approximation to $\min_A |S\Delta(A + A)|$?

The end!