# Recognizing Sumsets is NP-Complete

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

INSAIT
Research and Education Foundation

WEIZMANN INSTITUTE OF SCIENCE

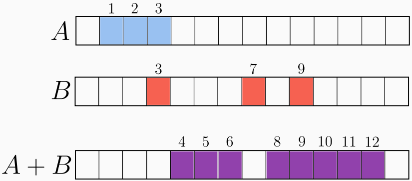
Funded by the European Union

erc
European Research Council
Established by the European Commission

# Sumsets

## Definition (Sum of two sets)

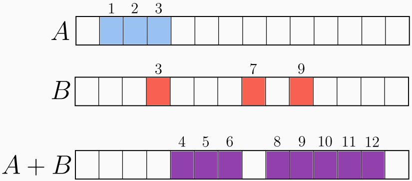
For $A, B \subseteq [0, M] := \{0, 1, \ldots, M\}$, let their sum be $A + B = \{a + b | a \in A, b \in B\}$.



# Sumsets

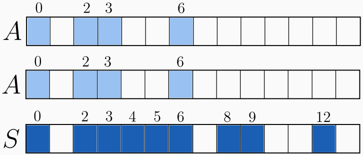
## Definition (Sum of two sets)

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 $A \subseteq [0, M]$.



# Sumset Recognition

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

# Sumset Recognition

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

**Example (Arithmetic Progressions)**

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

# Sumset Recognition

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

**Example (Arithmetic Progressions)**

If $A = [a, b]$, then $A + A = [2a, 2b]$, thus $|A + A| = 2|A| - 1$.
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 $|A + A| = \binom{|A|}{2} + |A|$.

# Sumset Recognition

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

**Example (Arithmetic Progressions)**

If $A = [a, b]$, then $A + A = [2a, 2b]$, thus $|A + A| = 2|A| - 1$.
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?

# Sumset Recognition

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

**Example (Arithmetic Progressions)**

If $A = [a, b]$, then $A + A = [2a, 2b]$, thus $|A + A| = 2|A| - 1$.
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?

**The Sumset Recognition Problem**

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

# Sumset Recognition

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

**Example (Arithmetic Progressions)**

If $A = [a, b]$, then $A + A = [2a, 2b]$, thus $|A + A| = 2|A| - 1$.
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?

**The Sumset Recognition Problem**

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

**Motivation:**

- Gain a better understanding of the structure sumsets.

# Sumset Recognition

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

**Example (Arithmetic Progressions)**

If $A = [a, b]$, then $A + A = [2a, 2b]$, thus $|A + A| = 2|A| - 1$.
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?

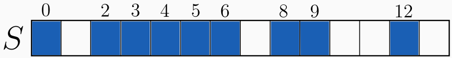
**The Sumset Recognition Problem**

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

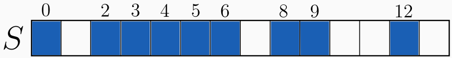
**Motivation:**

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

# Sumset Recognition Algorithms



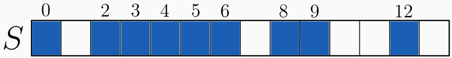
# Sumset Recognition Algorithms




**An $O^*(2^{M/2})$-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



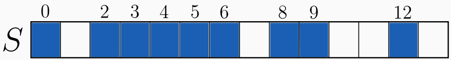

### An $O^*(2^{M/2})$-time algorithm

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

### An $O^*(2^n)$-time algorithm

Brute force only over feasible elements, i.e., all $x \in [0, M]$ such that $2x \in S$.

# Sumset Recognition Algorithms




**An $O^*(2^{M/2})$-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 $O^*(2^n)$-time algorithm**

Brute force only over feasible elements, i.e., all $x \in [0, M]$ 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 $\phi$ with $n$ variables, outputs a set $S \subseteq [0, O(n^4)]$, such that $S$ is a sumset if and only if $\phi$ 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 $\phi$ with $n$ variables, outputs a set $S \subseteq [0, O(n^4)]$, such that $S$ is a sumset if and only if $\phi$ is satisfiable.*

**Corollary**

Assuming the Exponential Time Hypothesis, there is no algorithm for Sumset Recognition running in time $2^{o(n^{1/4})}$.

# Our Contribution: Recognizing Sumsets is NP-Complete

**Theorem (Reduction from 3-SAT)**

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

**Corollary**

Assuming the Exponential Time Hypothesis, there is no algorithm for Sumset Recognition running in time $2^{o(n^{1/4})}$.

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

# Our Contribution: Recognizing Sumsets is NP-Complete

**Theorem (Reduction from 3-SAT)**

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

**Corollary**

Assuming the Exponential Time Hypothesis, there is no algorithm for Sumset Recognition running in time $2^{o(n^{1/4})}$.

**Open Question:** Resolve the gap between $2^{O(n)}$ and $2^{\Omega(n^{1/4})}$.
**Remark:** The results also apply for Sumset Recognition over $\mathbb{F}_p^d$ for any prime $p$.

# Proof Outline

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

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

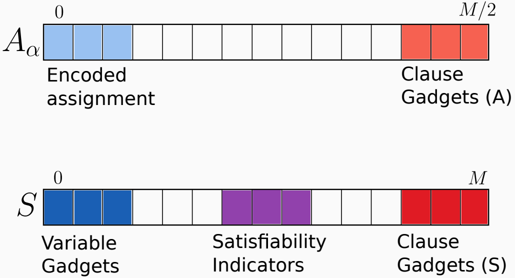
Moreover, $S = A_\alpha + A_\alpha$ for any satisfying assignment $\alpha \in \{0,1\}^n$.

# Proof Outline

We are given a 3-SAT formula $\phi$ 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 $\alpha \in \{0,1\}^n$.

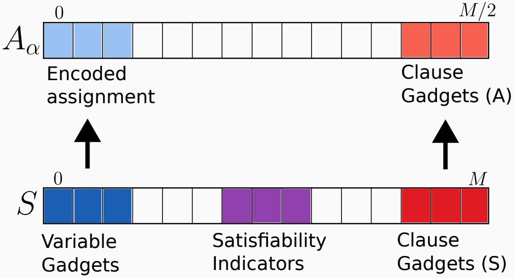


# Proof Outline

We are given a 3-SAT formula $\phi$ 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 $\alpha \in \{0,1\}^n$.

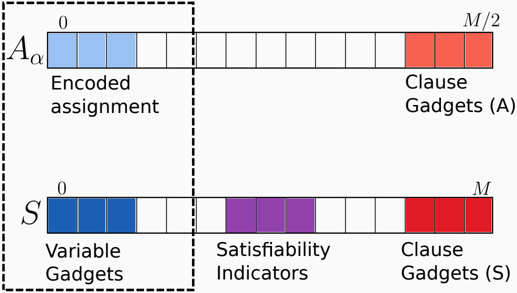


# Proof Outline

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

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

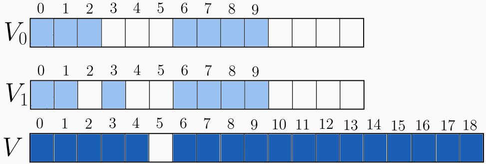
Moreover, $S = A_\alpha + A_\alpha$ for any satisfying assignment $\alpha \in \{0,1\}^n$.



# Variable Gadgets

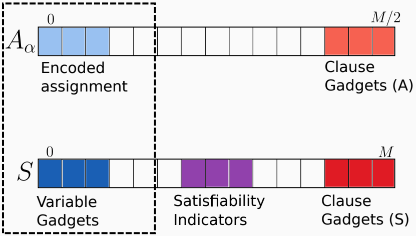
**Goal:** Design a constant sized set $V$ with only two representations: $V = V_0 + V_0$ and $V = V_1 + V_1$.

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

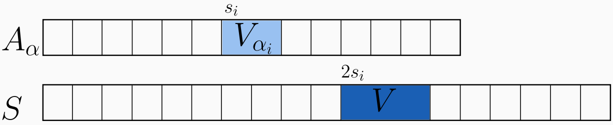



**Remark:** This is the smallest variable gadget.

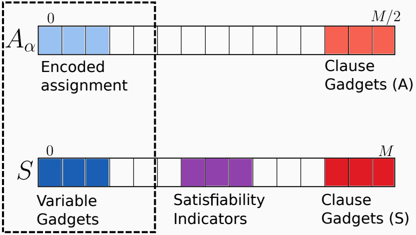
# Variable Gadgets



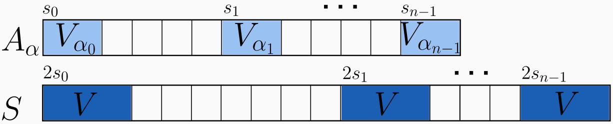

- Encoding one variable assignment at position $s_i$:



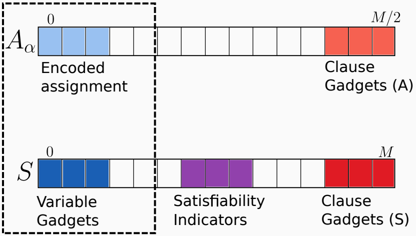
# Variable Gadgets



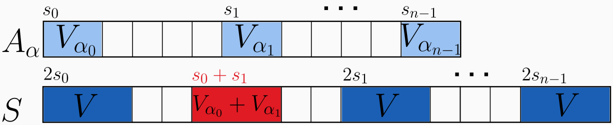

- Encoding one variable assignment at position $s_i$:
- Encoding $n$ variable assignments at positions $s_0 < \cdots < s_{n-1}$:



# Variable Gadgets

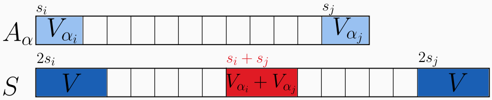



- Encoding one variable assignment at position $s_i$:
- Encoding $n$ variable assignments at positions $s_0 < \cdots < s_{n-1}$:



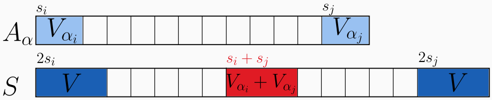

The cross-term $V_{\alpha_0} + V_{\alpha_1}$ is an unwanted by-product.

# 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:

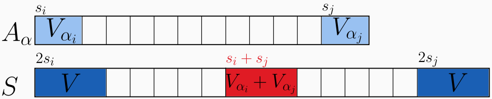
# Cross-Terms




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.

# Cross-Terms



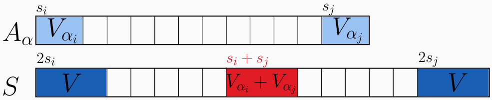

The cross-term $V_{\alpha_i} + V_{\alpha_j}$ appears at $s_i + s_j$ for every $i < j$. There are two problems with it:

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.

**Definition (Sidon Sets)**

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.

# 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:

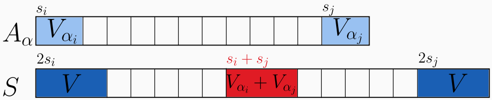
1. $s_i + s_j$ may collide with legitimate positions, i.e. $2s_k$. This problem can be solved by picking random $s_0, \ldots, s_{n-1}$, or deterministically, using *explicit* Sidon Sets.

**Definition (Sidon Sets)**

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.

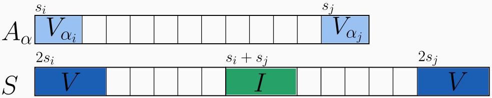
2. Having $V_{\alpha_i} + V_{\alpha_j}$ in $S$ will make $S$ dependent on $\alpha$. The set $S$ has to be *oblivious* to $\alpha$.

# Masking



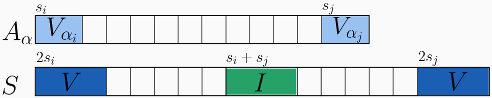

**Solution:** Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

# 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 $R + R = I$.

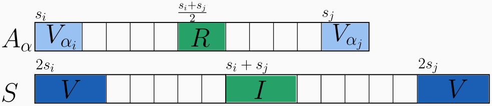
# Masking




**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 $A_\alpha$.

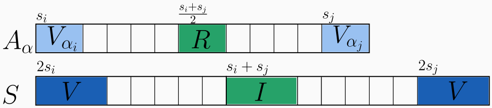
# Masking




**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 $A_\alpha$.

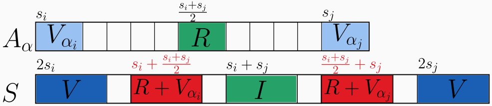
# Masking




**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 $A_\alpha$.
- New cross-terms: $R + V_{\alpha_i}$ and $R + V_{\alpha_j}$.

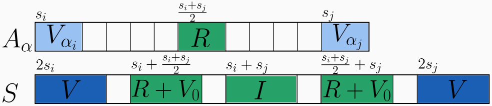
# Masking




**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 $A_\alpha$.
- New cross-terms: $R + V_{\alpha_i}$ and $R + V_{\alpha_j}$.

# Masking

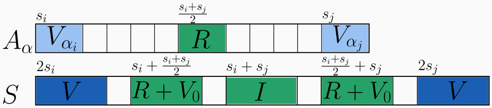



**Solution:** Mask the cross-terms with complete intervals. Let $I = [\min(V), \max(V)]$ and $R = [\frac{\min(V)}{2}, \frac{\max(V)}{2}]$, so $R + R = I$.

- Since $V_{\alpha_i} + V_{\alpha_j} \subset I$, $R$ has to appear in $A_\alpha$.
- 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 $\alpha$, so there's no "infinite game"!

# 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 $R + R = I$.

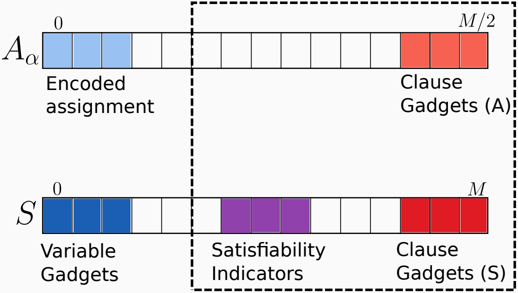
- Since $V_{\alpha_i} + V_{\alpha_j} \subset I$, $R$ has to appear in $A_\alpha$.
- New cross-terms: $R + V_{\alpha_i}$ and $R + V_{\alpha_j}$.

**Key idea:** $R + V_{\alpha_j}$ is an interval that's independent of $\alpha$, so there's no "infinite game"!

**Masking (informally)**

Whenever our design of $S$ and $A_\alpha$ fails because $A_\alpha + A_\alpha$ contains garbage terms, we can employ masking to fix it.

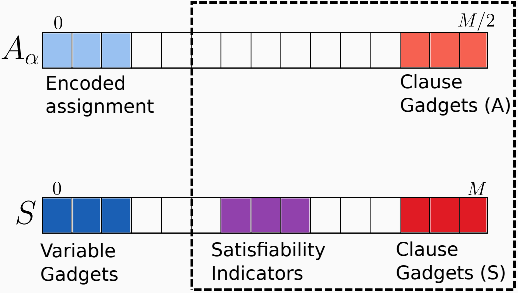
# Clause Gadgets




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

$$C_k \text{ is satisfied by } \alpha \iff (\text{encoded } \alpha) + G_k \ni t_k.$$

# Clause Gadgets




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

$$C_k \text{ is satisfied by } \alpha \iff (\text{encoded } \alpha) + G_k \ni t_k.$$

**Remark:** To enforce $G_k$ into $A_\alpha$, let $S$ contain $G_k + G_k$.

# Clause Gadgets

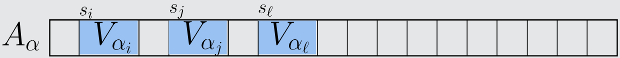
**Example (The gadget $G_k$)**

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

# Clause Gadgets

## Example (The gadget $G_k$)

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

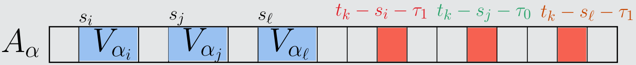



**Recall:** There are unique $\tau_0 \in V_0 \setminus V_1$ and $\tau_1 \in V_1 \setminus V_0$.

# Clause Gadgets

## Example (The gadget $G_k$)

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



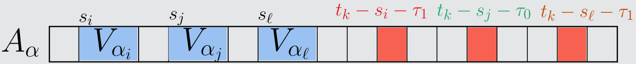

**Recall:** There are unique $\tau_0 \in V_0 \setminus V_1$ and $\tau_1 \in V_1 \setminus V_0$.
Let $t_k \gg s_n$, and set:

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

# 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 $t_k \gg s_n$, and set:

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

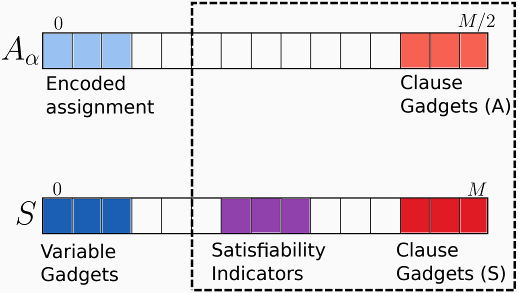
## Observation

$A_\alpha + A_\alpha$ 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 $\alpha_i$ satisfies the clause.

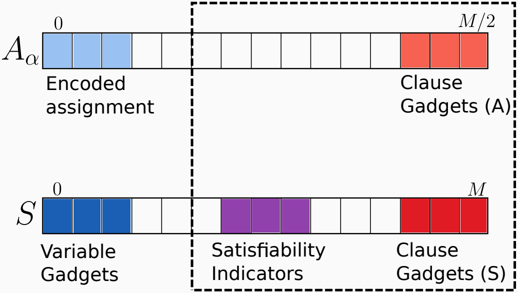
Hence, $t_k$ is in the set if and only if $\alpha$ satisfies the clause.

# Clause Gadgets



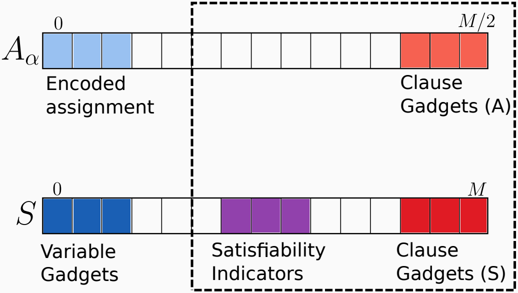

- Add $t_1 < t_2 < \ldots < t_m$ to $S$.

# Clause Gadgets




- Add $t_1 < t_2 < \ldots < t_m$ to $S$.
- For every $k \in [m]$, add $G_k$ to $A_\alpha$.

# Clause Gadgets




- Add $t_1 < t_2 < \ldots < t_m$ to $S$.
- For every $k \in [m]$, add $G_k$ to $A_\alpha$.

**Problem:** How could $S$ enforce $G_k$ into $A_\alpha$?

# Positioning

**More generally:** How could $S$ enforce some set $G$ into $A_\alpha$?

# Positioning

**More generally:** How could $S$ enforce some set $G$ into $A_\alpha$?
Having $G + G$ in $S$ fails because it could be that $G + G = R + R$ for some $R \neq G$.

# Positioning

**More generally:** How could $S$ enforce some set $G$ into $A_\alpha$?
Having $G+G$ in $S$ fails because it could be that $G+G=R+R$ for some $R \neq G$.

**Sumsets with a unique representation**

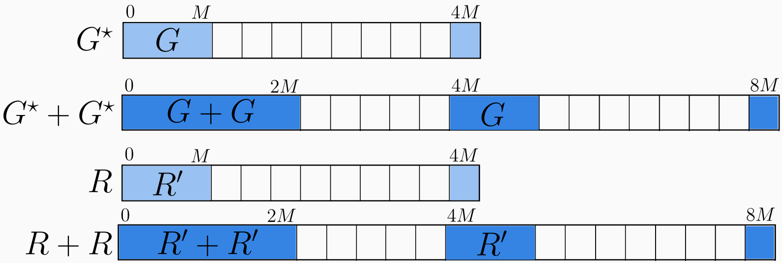
For any set $G \subseteq [0, M]$, the set $G^\star := G \cup \{4M\}$ is such that $G^\star + G^\star = R + R \iff R = G^\star$.

# Positioning

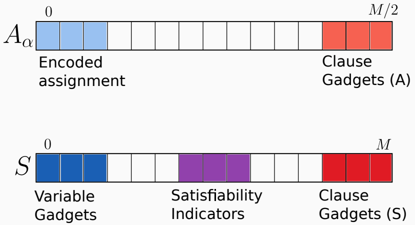
**More generally:** How could $S$ enforce some set $G$ into $A_\alpha$?
Having $G+G$ in $S$ fails because it could be that $G+G=R+R$ for some $R \neq G$.

**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, Positioning, and Wrapping it all Together




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

- Masking: Can ignore garbage terms in $A_\alpha + A_\alpha$.
- Positioning: Can enforce that $S = A + A$ if and only if $A = A_\alpha$ for some assignment $\alpha$.

# Open Questions

- Average-case complexity?
- Close the gap between $2^{\Omega(n^{1/4})}$ and $2^{O(n)}$?
- Approximation versions? E.g. $\min_A |S \Delta (A + A)|$.