Today

1) Bounding bad strings via compressions

2) Compressions from recursion trees

Recall

A k-SAT formula has overlap α if each clause shares variables w/ $\leq \alpha$ other clauses

$(x_1 \vee x_2) \wedge (x_3 \vee x_4)$ → overlap 0

$(x_1 \vee \bar{x}_2) \wedge (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (\bar{x}_3 \vee x_4)$ → overlap 3

Fact: Any k-SAT formula w/ $\alpha \leq \frac{2^k}{e} - 1$ is satisfiable

From Probabilistic Method to Algorithms

For LLL: Moser-Tardos Algorithm → works for LLL in general

MT

Assign $x_1, x_2, \ldots x_n$ UAR

While $\exists$ unsatisfied clause c

Fix(c)

Return $x_1, x_2, \ldots, x_n$

Fix(C)

Resample each $x_i \in C$

For each unsatisfied clause c' sharing variables w/ c → possibly c itself

Fix(c')

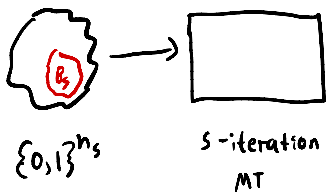
Analysis: "entropy compression" shows $O(m)$ iterations in $\mathbb{E}$

Theorem: If $\alpha \le 2^{k-c}$ for sufficiently large constant then Moser-Tardos finds a satisfying assignment in $O(m)$ iterations in expectation

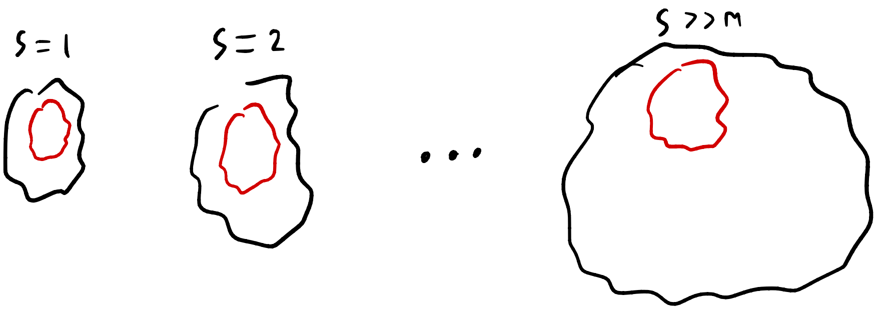
Intuition: Use execution of MT to compress bad bitstrings that make it take long

Let $\underline{n_s} := n + s\cdot k$ be the number of random bits used by s iterations of MT

Let $\underline{B_s} \subseteq \{0,1\}^{n_s}$ be all strings s.t. MT does not terminate after s iterations



Idea: Show B_s takes up vanishing fraction of n_s



Let $X :=$ iterations of MT

Suffices to show $\frac{|B_s|}{2^{n_s}} \le 2^{m-3s}$ b/c then

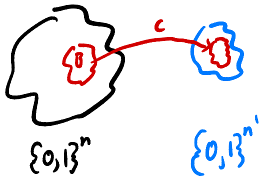
$$\mathbb{E}[X] = \sum_{s=0}^{\infty} s\cdot \Pr(X=s) \le m + \sum_{s\ge m}^{\infty} s\cdot \Pr(X=s) \le m + \sum_{s\ge m}^{\infty} s\cdot \Pr(X\ge s) \underset{\substack{\uparrow \\ B_s \text{ dfn.}}}{=} m + \sum_{s\ge m}^{\infty} s\cdot \frac{|B_s|}{2^{n_s}}$$

$$\le m + \sum_{s\ge m}^{\infty} s\cdot 2^{m-3s} \le m + \sum_{s\ge m}^{\infty} s\cdot 2^{-2s} \le m + \sum_{s\ge m}^{\infty} 2^{-s} = m + O(1)$$

Bounding $|B_s|$ via Compressions

Given a set $B \subseteq \{0,1\}^n$ a <u>compression</u> of B is an injective fn. $c: B \to \{0,1\}^{n'}$

↳ the <u>advantage</u> of c is $n - n'$



To get $\frac{|B_s|}{2^{n_s}} \leq 2^{m-3s}$ suffices to $\forall s$ give compression $C_s: B_s \to n_s'$ of advantage $3s-m$ b/c

$$\frac{|B_s|}{2^{n_s}} \underset{\substack{\uparrow \\ C_s \text{ injective}}}{\leq} \frac{2^{n_s'}}{2^{n_s}} = 2^{m-3s}$$

Will construct this compression using the "recursion tree" of MT

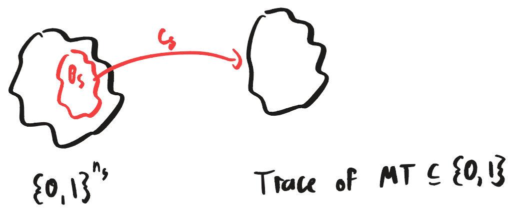
Compressions via Recursion Trees

Key idea: if Fix(C) called, learn all k bits of C

$\text{Fix}(x_1 \vee \bar{x}_2 \vee x_3) \to x_1 \vee \bar{x}_2 \vee x_3$ not satisfied $\to x_1 = 0, x_2 = 1, x_3 = 0$

but # possibilities for Fix(C') when Fix(C) called is $2^{k-c} \to$ needs only $k-c$ bits

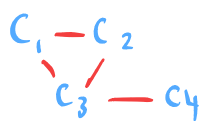
So $k \to k-c$ bits so compression



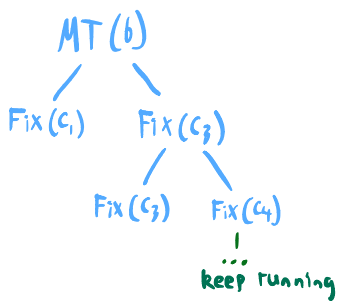
The recursion tree of MT(b) is the tree w/ root MT(b) and a node for each call of Fix w/ v child of u if u calls v

Formula: $\underbrace{(x_1 \vee \bar{x}_2)}_{C_1} \wedge \underbrace{(x_1 \vee x_2)}_{C_2} \wedge \underbrace{(x_1 \vee x_3)}_{C_3} \wedge \underbrace{(\bar{x}_3 \vee x_4)}_{C_4}$

Overlap:



Recursion Tree:



Let R_s be all recursion trees from running MT(b) for $b \in B_s$

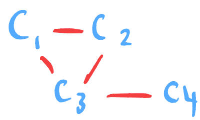
Lemma: Given recursion tree of MT(b) and $x_1, \dots, x_n$ after MT(b), can output b

$\left(\text{I.e. } \exists \text{ injective } f_s : B_s \to R_s \times \{0,1\}^n \right)$

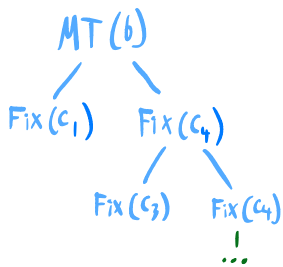
Proof by example

Formula: $\underbrace{(x_1 \lor \bar{x}_2)}_{C_1} \land \underbrace{(x_1 \lor x_2)}_{C_2} \land \underbrace{(x_1 \lor x_3)}_{C_3} \land \underbrace{(\bar{x}_3 \lor x_4)}_{C_4}$

Overlap:



Recursion Tree:



Reconstruction:

	n	sk	
b =	_ _ _ _ x_1 x_2 x_3 x_4	_ _ _ _ _ _ _	
			Fix(C_1)
	0 1 _ _ x_3 x_4	_ _ _ _ _ _ _ x_1 x_2	
			Fix(C_4)
	0 1 1 0	_ _ _ _ _ _ _ x_1 x_2 x_3 x_4	
			Fix(C_3)
	0 1 1 0	0 _ 0 _ _ _ _ x_2 x_4 x_1 x_3	
			Fix(C_4)
	0 1 1 0	0 _ 0 0 _ 1 _ _ $\underbrace{x_2 \quad x_1 \quad x_3 \; x_4}_{}$	

x_1, x_2, x_3, x_4 after running MT(b)

Lemma: Can describe a $T \in R_s$ w/ only $m + s(k-c+O(1))$ bits

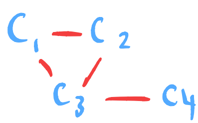
(I.e. $\exists$ injective $g_s : R \to \{0,1\}^{m+s(k-c+O(1))}$)

For adjacent C_i, C_j, define the index of C_j wrt C_i as i if j is ith smallest index of clause overlapping C_i

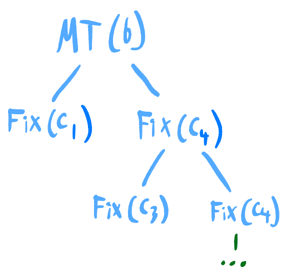
To describe $T \in R_s$:

- $\forall i$, 1 bit indicating if $\mathrm{Fix}(C_i)$ a child of $MT(b)$ → m bits
- For each $\mathrm{Fix}(C_i)$ → $s \cdot (k-c+O(1))$ bits
  - The index of each child of C_j → $k-c$ bits
  - $O(1)$ overhead for delimiters → $O(1)$ bits

Overlap:



Recursion Tree:



Description: $(1, 0, 0, 1 : (1, 2))$

1st neighbor 2nd neighbor

Final compression C_s is "composition" of f_s and g_s

for advantage $n+sk-(n+m+s(k-c+O(1))) \geq 3s - m$ for c sufficiently large