Construction and Applications of (k, d)-trees

Tibor Szabó

FU Berlin

November 5th, 2012

Coloring hypergraphs

k-uniform hypergraph $(X, \mathcal{F})$: X is a set, $\mathcal{F} \subseteq \binom{X}{k}$

Coloring hypergraphs

k-uniform hypergraph $(X, \mathcal{F})$: X is a set, $\mathcal{F} \subseteq \binom{X}{k}$

$\mathcal{F}$ is 2-colorable if there is a function (a "coloring")
$f : X \to \{$**red**, **blue**$\}$ such that no edge $A \in \mathcal{F}$ is monochromatic

Coloring hypergraphs

k-uniform hypergraph $(X, \mathcal{F})$: X is a set, $\mathcal{F} \subseteq \binom{X}{k}$

$\mathcal{F}$ is 2-colorable if there is a function (a "coloring")
$f : X \to \{$**red**, **blue**$\}$ such that no edge $A \in \mathcal{F}$ is monochromatic

Claim *If* $|\mathcal{F}| < 2^{k-1}$, *then* $\mathcal{F}$ *is* 2*-colorable.*

Coloring hypergraphs

k-uniform hypergraph $(X, \mathcal{F})$: X is a set, $\mathcal{F} \subseteq \binom{X}{k}$

$\mathcal{F}$ is 2-colorable if there is a function (a "coloring")
$f : X \to \{$**red**, **blue**$\}$ such that no edge $A \in \mathcal{F}$ is monochromatic

Claim *If* $|\mathcal{F}| < 2^{k-1}$, *then* $\mathcal{F}$ *is 2-colorable.*

Proof. Random 2-coloring. Color all $x \in X$ independently, uniformly:

$$Pr[f(x) = red] = \frac{1}{2} = Pr[f(x) = blue].$$

Coloring hypergraphs

k-uniform hypergraph $(X, \mathcal{F})$: X is a set, $\mathcal{F} \subseteq \binom{X}{k}$

$\mathcal{F}$ is 2-colorable if there is a function (a "coloring")
$f : X \to \{$**red**, **blue**$\}$ such that no edge $A \in \mathcal{F}$ is monochromatic

Claim *If* $|\mathcal{F}| < 2^{k-1}$, *then* $\mathcal{F}$ *is 2-colorable.*

Proof. Random 2-coloring. Color all $x \in X$ independently, uniformly:

$$Pr[f(x) = \mathit{red}] = \frac{1}{2} = Pr[f(x) = \mathit{blue}].$$

For $A \in \mathcal{F}$, let $Y_A = 1$ if A is monochromatic, otherwise $Y_A = 0$.

$$\mathbb{E}[\#\text{of m.c. edges of } \mathcal{F}] = \mathbb{E}\left[\sum_{A \in \mathcal{F}} Y_A\right] = \sum_{A \in \mathcal{F}} \mathbb{E} Y_A = \frac{|\mathcal{F}|}{2^{k-1}} < 1$$

Coloring hypergraphs

k-uniform hypergraph $(X, \mathcal{F})$: X is a set, $\mathcal{F} \subseteq \binom{X}{k}$

$\mathcal{F}$ is 2-colorable if there is a function (a "coloring")
$f : X \to \{$**red**, **blue**$\}$ such that no edge $A \in \mathcal{F}$ is monochromatic

Claim *If* $|\mathcal{F}| < 2^{k-1}$, *then* $\mathcal{F}$ *is 2-colorable.*

Proof. Random 2-coloring. Color all $x \in X$ independently, uniformly:

$$Pr[f(x) = red] = \frac{1}{2} = Pr[f(x) = blue].$$

For $A \in \mathcal{F}$, let $Y_A = 1$ if A is monochromatic, otherwise $Y_A = 0$.

$$\mathbb{E}[\#\text{of m.c. edges of } \mathcal{F}] = \mathbb{E}\left[\sum_{A \in \mathcal{F}} Y_A\right] = \sum_{A \in \mathcal{F}} \mathbb{E} Y_A = \frac{|\mathcal{F}|}{2^{k-1}} < 1$$

Hence, for sure, **THERE EXISTS** 2-coloring without m.c. edges ("proper 2-coloring") $\square$

Question: Is there an (efficient, deterministic) algorithm which *finds* a proper 2-coloring?

Question: Is there an (efficient, deterministic) algorithm which *finds* a proper 2-coloring?

YES!

Question: Is there an (efficient, deterministic) algorithm which *finds* a proper 2-coloring?

YES!

$\rightsquigarrow$ Positional games

Positional Games

Maker-Breaker Game $(X, \mathcal{F})$:

Board: set X; family of **winning sets**: $\mathcal{F} \subset 2^X$

Players: Maker and Breaker

Play: players alternately occupy elements of X; Maker starts

Winner: Maker if he occupies a winning set completely

Breaker, if he puts his mark in every winning set

Maker-Breaker Game $(X, \mathcal{F})$:

Board: set X; family of **winning sets**: $\mathcal{F} \subset 2^X$

Players: Maker and Breaker

Play: players alternately occupy elements of X; Maker starts

Winner: Maker if he occupies a winning set completely

Breaker, if he puts his mark in every winning set

Remark: Perfect information game with complementary goals:
1) Exactly one of the players has a winning strategy.
2) Given $\mathcal{F}$, it is clear (at least to an all-powerful computer) which of them has a winning strategy.

Terminology: $\mathcal{F}$ is Maker's win, $\mathcal{F}$ is Breaker's win

Maker-Breaker Game $(X, \mathcal{F})$:

Board: set X; family of **winning sets**: $\mathcal{F} \subset 2^X$

Players: Maker and Breaker

Play: players alternately occupy elements of X; Maker starts

Winner: Maker if he occupies a winning set completely
Breaker, if he puts his mark in every winning set

Remark: Perfect information game with complementary goals:
1) Exactly one of the players has a winning strategy.
2) Given $\mathcal{F}$, it is clear (at least to an all-powerful computer) which of them has a winning strategy.
Terminology: $\mathcal{F}$ is Maker's win, $\mathcal{F}$ is Breaker's win

Proposition: $\mathcal{F}$ is a Breaker's win $\Rightarrow$ $\mathcal{F}$ is 2-colorable

Maker-Breaker Game $(X, \mathcal{F})$:

Board: set X; family of **winning sets**: $\mathcal{F} \subset 2^X$

Players: Maker and Breaker

Play: players alternately occupy elements of X; Maker starts

Winner: Maker if he occupies a winning set completely
Breaker, if he puts his mark in every winning set

Remark: Perfect information game with complementary goals:
1) Exactly one of the players has a winning strategy.
2) Given $\mathcal{F}$, it is clear (at least to an all-powerful computer) which of them has a winning strategy.
Terminology: $\mathcal{F}$ is Maker's win, $\mathcal{F}$ is Breaker's win

Proposition: $\mathcal{F}$ is a Breaker's win $\Rightarrow$ $\mathcal{F}$ is 2-colorable

Erdős-Selfridge: $|\mathcal{F}| < 2^{k-1} \Rightarrow$ Breaker has a winning strategy.

Lovász Local Lemma – Neighborhood Conjecture

LLL. $A_1, A_2, \ldots, A_k$ events in some probability space, such that
(1) every A_i is mutually independent from **all but** d other events
(2) $p \geq Pr[A_i]$ for every i
If $ep(d+1) \leq 1$ then $Pr[\wedge_{i=1}^{k} \overline{A_i}] > 0$.

Lovász Local Lemma – Neighborhood Conjecture

LLL. $A_1, A_2, \dots, A_k$ events in some probability space, such that
(1) every A_i is mutually independent from **all but** d other events
(2) $p \geq Pr[A_i]$ for every i
If $ep(d+1) \leq 1$ then $Pr[\wedge_{i=1}^k \overline{A_i}] > 0$.

Application of LLL: $\Delta(L(\mathcal{F})) \leq \frac{2^{k-1}}{e} - 1 \;\Rightarrow\; \mathcal{F}$ is 2-colorable.
$\Delta(\mathcal{F}) \leq \frac{2^{k-1}}{ek} \;\Rightarrow\; \mathcal{F}$ is 2-colorable.

LLL. $A_1, A_2, \ldots, A_k$ events in some probability space, such that
(1) every A_i is mutually independent from **all but** d other events
(2) $p \geq Pr[A_i]$ for every i
If $ep(d+1) \leq 1$ then $Pr[\wedge_{i=1}^{k} \overline{A_i}] > 0$.

Application of LLL: $\Delta(L(\mathcal{F})) \leq \frac{2^{k-1}}{e} - 1 \;\Rightarrow\; \mathcal{F}$ is 2-colorable.
$\Delta(\mathcal{F}) \leq \frac{2^{k-1}}{ek} \;\Rightarrow\; \mathcal{F}$ is 2-colorable.

Moser-Tardos (2010) There is a (simple) algorithm to efficiently 2-color such an $\mathcal{F}$.

Lovász Local Lemma – Neighborhood Conjecture

LLL. $A_1, A_2, \ldots, A_k$ events in some probability space, such that
(1) every A_i is mutually independent from **all but** d other events
(2) $p \geq Pr[A_i]$ for every i
If $ep(d+1) \leq 1$ then $Pr[\wedge_{i=1}^{k} \overline{A_i}] > 0$.

Application of LLL: $\Delta(L(\mathcal{F})) \leq \frac{2^{k-1}}{e} - 1 \;\Rightarrow\; \mathcal{F}$ is 2-colorable.
$\Delta(\mathcal{F}) \leq \frac{2^{k-1}}{ek} \;\Rightarrow\; \mathcal{F}$ is 2-colorable.

Moser-Tardos (2010) There is a (simple) algorithm to efficiently 2-color such an $\mathcal{F}$.

Neighborhood Conjecture [Beck]

$$\Delta(L(\mathcal{F})) < 2^{k-1} \;\Rightarrow\; \mathcal{F} \text{ is Breaker's win.}$$

Counterexamples to the strongest form of NC

Theorem (Gebauer, '09)

(i) For every large enough k, there is a k-uniform Maker's win hypergraph $\mathcal{H}$ with $\Delta(L(\mathcal{H})) \leq 0.75 \cdot 2^{k-1}$

(ii) For every large enough k there is a k-uniform Maker's win hypergraph $\mathcal{F}$ with $\Delta(\mathcal{F}) < 0.5 \cdot \frac{2^k}{k}$.

$$D(k) := \min\{\Delta(\mathcal{F}) : \ k\text{-uniform, Maker's win } \mathcal{F}\}$$

Best know lower bound $D(k) > \lfloor \frac{k}{2} \rfloor$.
Deciding whether $D(k) = \lfloor \frac{k}{2} \rfloor + 1$ already seems to need new ideas.

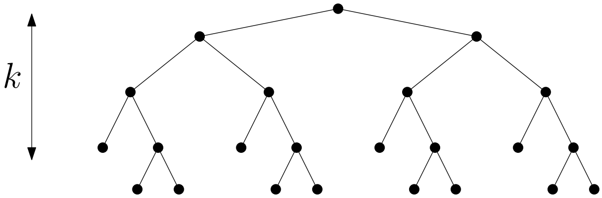
Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

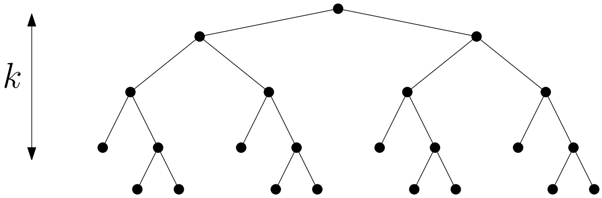
$k = 3:$



Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

$k = 3$:

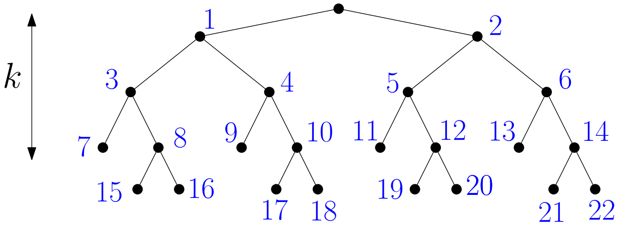


vertices $\leftrightsquigarrow$ elements of the board X

Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

$k = 3$:

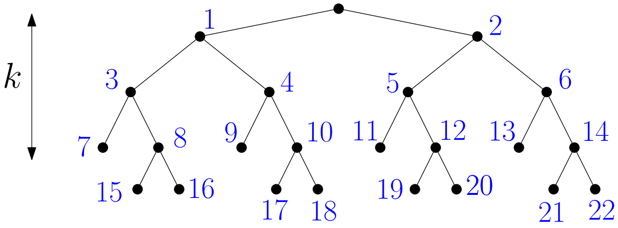


vertices $\leftrightsquigarrow$ elements of the board X

Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

$k = 3:$



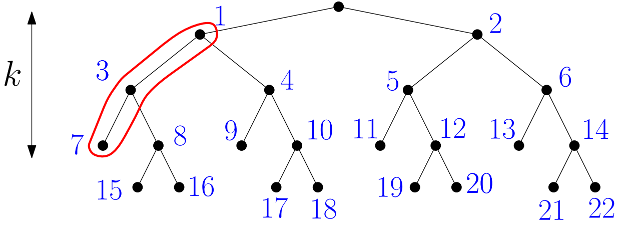
vertices $\leftrightsquigarrow$ elements of the board X

winning sets $\leftrightsquigarrow$ end-paths

Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

$k = 3$:



$\mathcal{F} := \{\{1, 3, 7\}\}$

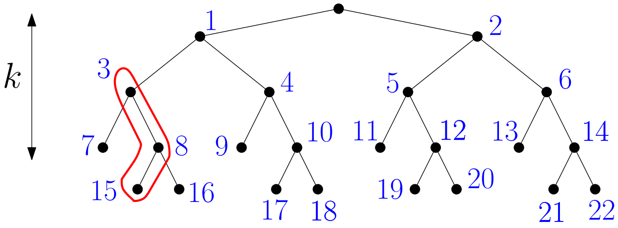
vertices $\leftrightsquigarrow$ elements of the board X

winning sets $\leftrightsquigarrow$ end-paths

Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

$k = 3:$



$\mathcal{F} := \{\{1, 3, 7\}, \{3, 8, 15\}\}$

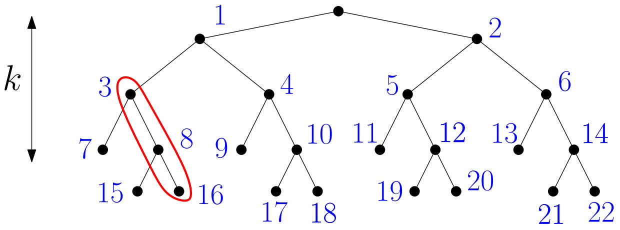
vertices $\leftrightsquigarrow$ elements of the board X

winning sets $\leftrightsquigarrow$ end-paths

Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

$k = 3:$



$\mathcal{F} := \{\{1, 3, 7\}, \{3, 8, 15\}, \{3, 8, 16\}\}$

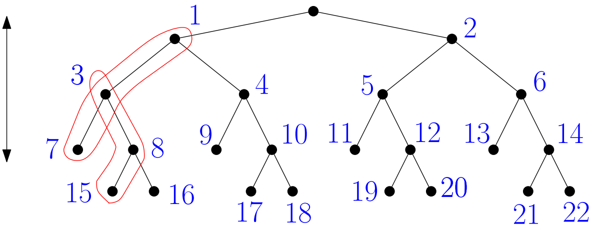
vertices $\leftrightsquigarrow$ elements of the board X

winning sets $\leftrightsquigarrow$ end-paths

Game-hypergraphs from trees

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

$k = 3:$



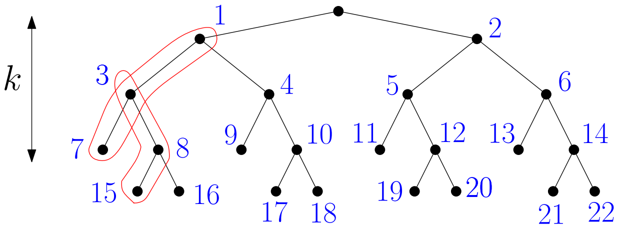
$\mathcal{F} := \{\{1,3,7\},\{3,8,15\},\{3,8,16\},\ldots\}$

vertices $\leftrightsquigarrow$ elements of the board X

winning sets $\leftrightsquigarrow$ end-paths

full binary trees $\rightsquigarrow$ k-uniform hypergraphs

$k = 3:$



$\mathcal{F} := \{\{1,3,7\},\{3,8,15\},\{3,8,16\},\ldots\}$

vertices $\leftrightsquigarrow$ elements of the board X

winning sets $\leftrightsquigarrow$ end-paths

Proposition

Maker has a winning strategy on the hypergraph $\mathcal{F}$.

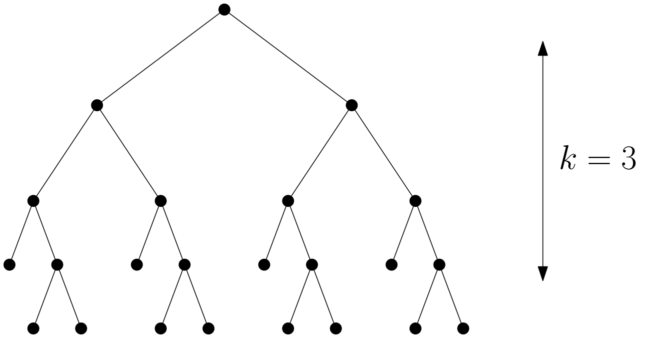
Def. (k,d)-tree

- Every leaf has depth $\geq k$
- Every vertex has $\leq d$ leaf-descendants of distance $\leq k$

Def. (k, d)-tree

- Every leaf has depth $\geq k$
- Every vertex has $\leq d$ leaf-descendants of distance $\leq k$

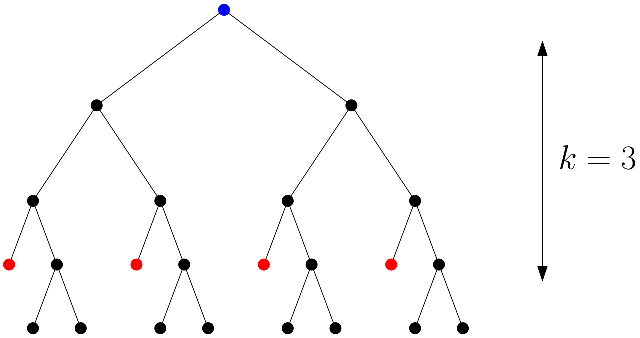
A $(3, 6)$-tree



Def. (k, d)-tree

- Every leaf has depth $\geq k$
- Every vertex has $\leq d$ leaf-descendants of distance $\leq k$

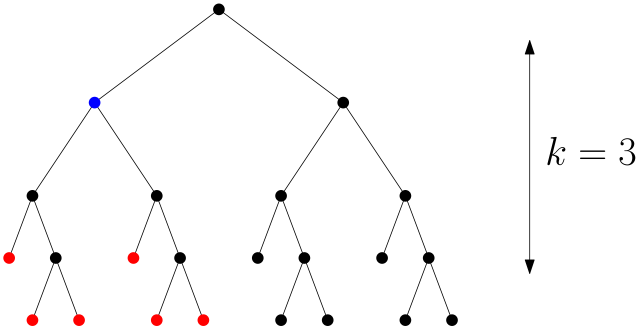
A $(3, 6)$-tree



Def. (k, d)-tree

- Every leaf has depth $\geq k$
- Every vertex has $\leq d$ leaf-descendants of distance $\leq k$

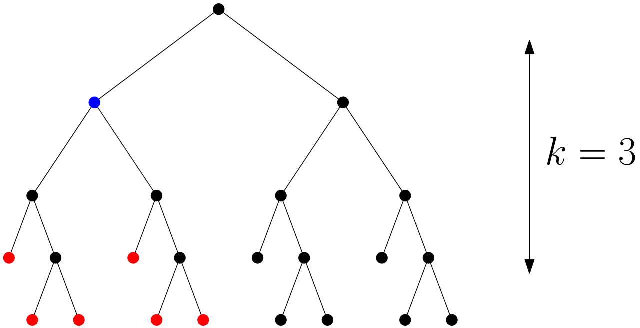
A $(3, 6)$-tree



Def. (k, d)-tree

- Every leaf has depth $\geq k$
- Every vertex has $\leq d$ leaf-descendants of distance $\leq k$

A $(3, 6)$-tree



Proposition

There is a $(k-1, d)$-tree $\Rightarrow$ $D(k) \leq d$

Constructing (k, d)-trees

Theorem (Gebauer - Sz. - Tardos, 2011)

There exists a (k, d)-tree with

$$d = \left(\frac{2}{e} + o(1)\right)\frac{2^k}{k}.$$

Corollary

For every positive integer k there exists Maker's win k-uniform hypergraphs $\mathcal{H}$ and $\mathcal{H}'$, such that

(i) $\Delta(L(\mathcal{H})) = \left(1 + O\left(\frac{1}{\sqrt{k}}\right)\right)\frac{2^{k-1}}{e}$,

(ii) $\Delta(\mathcal{H}) = \left(1 + O\left(\frac{1}{\sqrt{k}}\right)\right)\frac{2^k}{ek}$.

Application 2 of LLL: (k,s)-SAT

Application 2 of LLL: Let F be a boolean CNF-formula such that every clause contains exactly k distinct literals. If every variable occurs in less than $\frac{1}{e} \cdot \frac{2^k}{k}$, then F is satisfiable.

Application 2 of LLL: (k,s)-SAT

Application 2 of LLL: Let F be a boolean CNF-formula such that every clause contains exactly k distinct literals. If every variable occurs in less than $\frac{1}{e} \cdot \frac{2^k}{k}$, then F is satisfiable.

Def. (k, s)-CNF formula

- every clause contains exactly k distinct literals
- every variable occurs in at most s clauses

Application 2 of LLL: (k,s)-SAT

Application 2 of LLL: Let F be a boolean CNF-formula such that every clause contains exactly k distinct literals. If every variable occurs in less than $\frac{1}{e} \cdot \frac{2^k}{k}$, then F is satisfiable.

Def. (k, s)-CNF formula

- every clause contains exactly k distinct literals
- every variable occurs in at most s clauses

Extremal question: How large is

$$f(k) := \max\{s : \text{every } (k, s)\text{-SAT is satisfiable}\}??$$

Application 2 of LLL: (k,s)-SAT

Application 2 of LLL: Let F be a boolean CNF-formula such that every clause contains exactly k distinct literals. If every variable occurs in less than $\frac{1}{e} \cdot \frac{2^k}{k}$, then F is satisfiable.

Def. (k, s)-CNF formula

- every clause contains exactly k distinct literals
- every variable occurs in at most s clauses

Extremal question: How large is

$$f(k) := \max\{s : \text{every } (k, s)\text{-SAT is satisfiable}\}??$$

Known values: $f(3) = 3$, $f(4) = 4$, $f(5) = ?$
f is NOT known to be computable

Upper bounds:

	$k \cdot \frac{2^k}{k}$	trivial
	$k^{0.74} \cdot \frac{2^k}{k}$	Savicky-Sgall, '00
	$\log k \cdot \frac{2^k}{k}$	Hoory-Szeider, '06
	$1 \cdot \frac{2^k}{k}$	Gebauer, '09

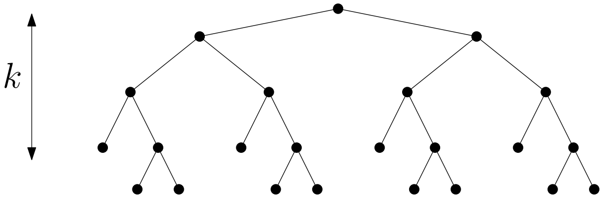
Formulas from trees

full binary trees $\rightsquigarrow$ k-CNF formulas

Formulas from trees

full binary trees $\rightsquigarrow$ k-CNF formulas

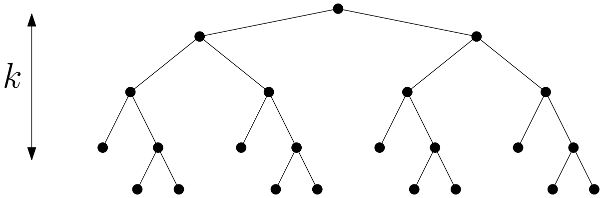
$k = 3:$



Formulas from trees

full binary trees $\rightsquigarrow$ k-CNF formulas

$k = 3:$

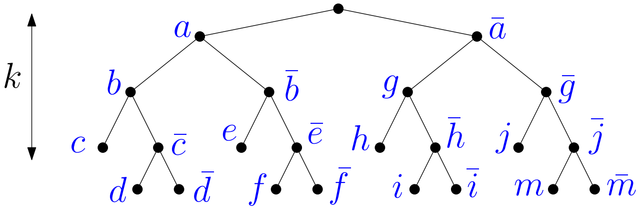


vertices $\leftrightsquigarrow$ distinct literals
siblings $\leftrightsquigarrow$ opposite literals

Formulas from trees

full binary trees $\rightsquigarrow$ k-CNF formulas

$k = 3$:



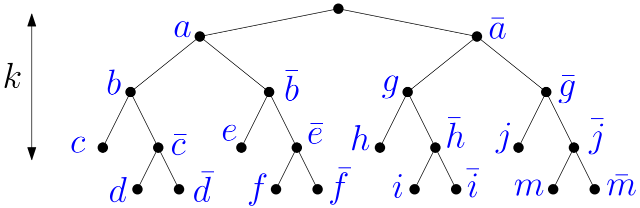
vertices $\leftrightsquigarrow$ distinct literals

siblings $\leftrightsquigarrow$ opposite literals

Formulas from trees

full binary trees $\rightsquigarrow$ k-CNF formulas

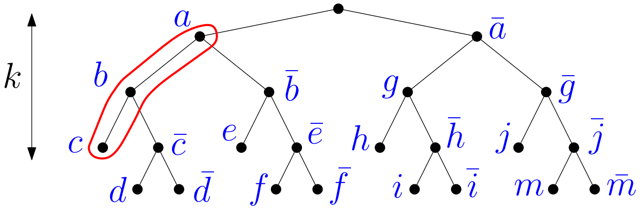
$k = 3:$



vertices $\leftrightsquigarrow$ distinct literals
siblings $\leftrightsquigarrow$ opposite literals
clauses $\leftrightsquigarrow$ end-paths

full binary trees $\rightsquigarrow$ k-CNF formulas

$k = 3:$



$F = (c \vee b \vee a)$

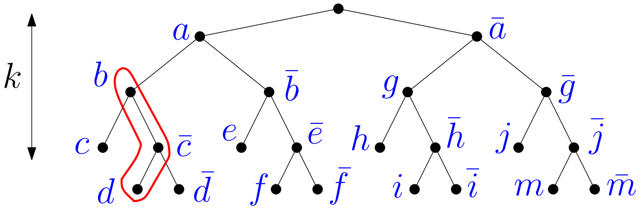
vertices $\leftrightsquigarrow$ distinct literals

siblings $\leftrightsquigarrow$ opposite literals

clauses $\leftrightsquigarrow$ end-paths

full binary trees $\rightsquigarrow$ k-CNF formulas

$k = 3$:



$F = (c \vee b \vee a) \wedge (d \vee \bar{c} \vee b)$

vertices $\leftrightsquigarrow$ distinct literals

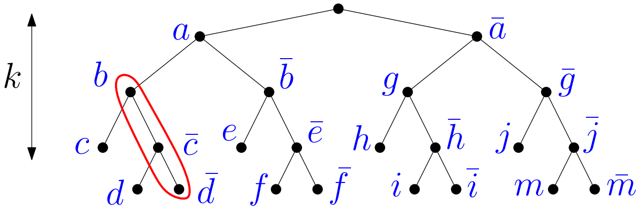
siblings $\leftrightsquigarrow$ opposite literals

clauses $\leftrightsquigarrow$ end-paths

Formulas from trees

full binary trees $\rightsquigarrow$ k-CNF formulas

$k = 3$:



$F = (c \vee b \vee a) \wedge (d \vee \bar{c} \vee b) \wedge (\bar{d} \vee \bar{c} \vee b)$

vertices $\leftrightsquigarrow$ distinct literals

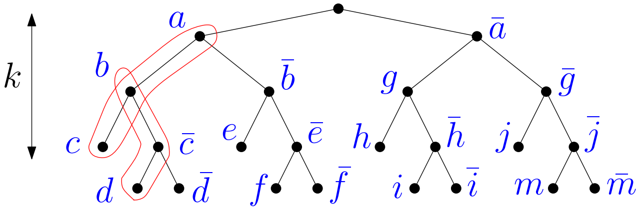
siblings $\leftrightsquigarrow$ opposite literals

clauses $\leftrightsquigarrow$ end-paths

Formulas from trees

full binary trees $\rightsquigarrow$ k-CNF formulas

$k = 3:$



$F = (c \vee b \vee a) \wedge (d \vee \bar{c} \vee b) \wedge (\bar{d} \vee \bar{c} \vee b) \wedge \ldots$

vertices $\leftrightsquigarrow$ distinct literals

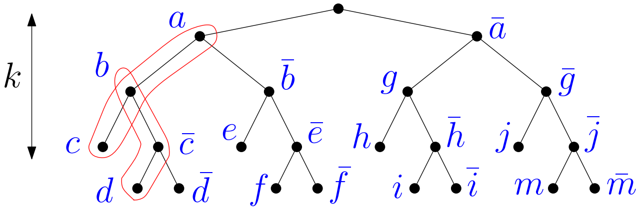
siblings $\leftrightsquigarrow$ opposite literals

clauses $\leftrightsquigarrow$ end-paths

Formulas from trees

full binary trees $\rightsquigarrow$ k-CNF formulas

$k = 3:$



$F = (c \vee b \vee a) \wedge (d \vee \bar{c} \vee b) \wedge (\bar{d} \vee \bar{c} \vee b) \wedge \ldots$

vertices $\leftrightsquigarrow$ distinct literals

siblings $\leftrightsquigarrow$ opposite literals

clauses $\leftrightsquigarrow$ end-paths

Proposition

The obtained formula is NOT satisfiable

Asymptotics via (k, d)-trees

Proposition

There is a (k, d)*-tree* $\quad \Rightarrow \quad f(k) < d$

Asymptotics via (k, d)-trees

Proposition

There is a (k, d)-tree $\Rightarrow$ $f(k) < d$

Theorem (Gebauer-Sz.-Tardos, '11)

$$f(k) = \left(\frac{2}{e} + o(1)\right) \frac{2^k}{k}$$

Asymptotics via (k, d)-trees

Proposition

There is a (k, d)-tree $\Rightarrow$ $f(k) < d$

Theorem (Gebauer-Sz.-Tardos, '11)

$$f(k) = \left(\frac{2}{e} + o(1)\right)\frac{2^k}{k}$$

Def. The degree of clause C in formula F is the number of those clauses of F that share a variable with C.
$D(F)$ denotes the largest of all clause-degrees in F.

$l(k) := \max\{s : \text{ every } k\text{-CNF formula } F \text{ with } D(F) \leq s \text{ is saitisfiable}\}$

Asymptotics via (k, d)-trees

Proposition

There is a (k, d)-tree $\Rightarrow$ $f(k) < d$

Theorem (Gebauer-Sz.-Tardos, '11)

$$f(k) = \left(\frac{2}{e} + o(1)\right)\frac{2^k}{k}$$

Def. The degree of clause C in formula F is the number of those clauses of F that share a variable with C.
$D(F)$ denotes the largest of all clause-degrees in F.

$l(k) := \max\{s : \text{ every } k\text{-CNF formula } F \text{ with } D(F) \leq s \text{ is saitisfiable}\}$

Theorem (Gebauer-Sz.-Tardos, '11)

$$l(k) = \left(\frac{1}{e} + o(1)\right)2^k$$

Complexity hardness jump

(k, s)-SAT Problem

- Input: a (k, s)-CNF F
- Decide whether F is satisfiable

Complexity hardness jump

(k, s)-SAT Problem

- Input: a (k, s)-CNF F
- Decide whether F is satisfiable

Hardness Jump [Tovey '84; Kratochvíl-Savický-Tuza '93]

Complexity hardness jump

(k,s)-SAT Problem

- Input: a (k,s)-CNF F
- Decide whether F is satisfiable

Hardness Jump [Tovey '84; Kratochvíl-Savický-Tuza '93]

- $(k,1)$-SAT trivial
- $(k,2)$-SAT trivial

$\vdots$ $\vdots$

- $(k, f(k))$-SAT trivial

Complexity hardness jump

(k, s)-SAT Problem

- Input: a (k, s)-CNF F
- Decide whether F is satisfiable

Hardness Jump [Tovey '84; Kratochvíl-Savický-Tuza '93]

- $(k, 1)$-SAT trivial
- $(k, 2)$-SAT trivial

$\vdots$ $\vdots$

- $(k, f(k))$-SAT trivial

. . . .

- $(k, f(k) + 1)$-SAT NP-hard

$\vdots$ $\vdots$

- (k, ∞)-SAT NP-hard

Improved lower bound with Lopsided LLL

For LLL: set every variable x to $\texttt{true}$ with probability $P_x = \frac{1}{2}$.
Works for every (k, s)-CNF formula F with $s = \left\lfloor \frac{1}{e} \cdot \frac{2^k}{k} \right\rfloor$.

Improved lower bound with Lopsided LLL

For LLL: set every variable x to `true` with probability $P_x = \frac{1}{2}$.
Works for every (k, s)-CNF formula F with $s = \left\lfloor \frac{1}{e} \cdot \frac{2^k}{k} \right\rfloor$.

How to improve? Take the particular formula F into account
[idea of Berman-Karpinski-Scott, '04]
LLLL: Cares only about conflicting occurrences of variables

Lemma

(Lopsided Local Lemma) Let $\{A_C\}_{C\in I}$ *be a finite set of events in some probability space. Let* $\Gamma(C)$ *be a subset of* I *for each* $C \in I$ *such that for every subset* $J \subseteq I \setminus (\Gamma(C) \cup \{C\})$ *we have*

$$Pr(A_C | \wedge_{D\in J} \bar{A}_D) \leq Pr(A_C).$$

Suppose there are real numbers $0 < x_C < 1$ *for* $C \in I$ *such that for every* $C \in I$ *we have*

$$Pr(A_C) \leq x_C \prod_{D\in\Gamma(C)} (1 - x_D).$$

Then

$$Pr(\wedge_{C\in I} \bar{A}_C) > 0.$$

Improved lower bound with Lopsided LLL

For LLL: set every variable x to $\texttt{true}$ with probability $P_x = \frac{1}{2}$.
Works for every (k, s)-CNF formula F with $s = \left\lfloor \frac{1}{e} \cdot \frac{2^k}{k} \right\rfloor$.

How to improve? Take the particular formula F into account
[idea of Berman-Karpinski-Scott, '04]
LLLL: Cares only about conflicting occurrences of variables

Improved lower bound with Lopsided LLL

For LLL: set every variable x to $\mathtt{true}$ with probability $P_x = \frac{1}{2}$.
Works for every (k, s)-CNF formula F with $s = \left\lfloor \frac{1}{e} \cdot \frac{2^k}{k} \right\rfloor$.

How to improve? Take the particular formula F into account
[idea of Berman-Karpinski-Scott, '04]
LLLL: Cares only about conflicting occurrences of variables

Set variable x to $\mathtt{true}$ with probability $P_x = \frac{1}{2} + \frac{2d_{\bar{x}} - s}{2sk}$,
where for literal v let $d_v :=$ # of occurrences of v in F.

Improved lower bound with Lopsided LLL

For LLL: set every variable x to $\texttt{true}$ with probability $P_x = \frac{1}{2}$.
Works for every (k,s)-CNF formula F with $s = \left\lfloor \frac{1}{e} \cdot \frac{2^k}{k} \right\rfloor$.

How to improve? Take the particular formula F into account
[idea of Berman-Karpinski-Scott, '04]
LLLL: Cares only about conflicting occurrences of variables

Set variable x to $\texttt{true}$ with probability $P_x = \frac{1}{2} + \frac{2d_{\bar{x}} - s}{2sk}$,
where for literal v let $d_v := \#$ of occurrences of v in F.
Surprise: The *more* a variable appears in the clauses of F as non-negated, *the less likely* we will set it to true.

Improved lower bound with Lopsided LLL

For LLL: set every variable x to `true` with probability $P_x = \frac{1}{2}$.
Works for every (k,s)-CNF formula F with $s = \left\lfloor \frac{1}{e} \cdot \frac{2^k}{k} \right\rfloor$.

How to improve? Take the particular formula F into account
[idea of Berman-Karpinski-Scott, '04]
LLLL: Cares only about conflicting occurrences of variables

Set variable x to `true` with probability $P_x = \frac{1}{2} + \frac{2d_{\bar{x}} - s}{2sk}$,
where for literal v let $d_v := \#$ of occurrences of v in F.
Surprise: The *more* a variable appears in the clauses of F as non-negated, *the less likely* we will set it to true.

Works for every (k,s)-CNF formula F with $s = \left\lfloor \frac{2}{e} \cdot \frac{2^k}{k+1} \right\rfloor$.

Liar Game Player A thinks of an integer $x \in [N]$ and Player B tries to figure it out by asking Yes/No questions of the sort "Is $x \in S$?", where S is a subset of $[N]$ picked by B.

A is allowed to lie. However for B to have a chance to be successful, but the lies have to come in some controlled fashion.

Liar Game Player A thinks of an integer $x \in [N]$ and Player B tries to figure it out by asking Yes/No questions of the sort "Is $x \in S$?", where S is a subset of $[N]$ picked by B.

A is allowed to lie. However for B to have a chance to be successful, but the lies have to come in some controlled fashion.

Ulam's problem for binary search with k lies: A is allowed to lie a total of k times What is the smallest number $q(N, k)$ of questions that allows B to figure out the answer.

Instead of limiting the total number of lies, now the number of consecutive lies is limited: A is not allowed to lie k consecutive times

This restriction on the lies is not enough for B to find the value x with certainty, but he will be able to narrow the set of possibilities.

The IMO problem asked for estimates on how small B can guarantee this set of possibilities will eventually be.

Problem 3, 2012 International Mathematics Olympiad

Instead of limiting the total number of lies, now the number of consecutive lies is limited: A is not allowed to lie k consecutive times

This restriction on the lies is not enough for B to find the value x with certainty, but he will be able to narrow the set of possibilities. The IMO problem asked for estimates on how small B can guarantee this set of possibilities will eventually be.

Theorem

(Gebauer-Sz.-Tardos) Let $N > d$ and k be positive integers. Assume A and B play the game in which A thinks of an element $x \in [N]$ and then answers an arbitrary number of B's questions of the form "Is $x \in S$?". Assume further that A is allowed to lie, but never to k consecutive questions. Then B can guarantee to narrow the number of possibilities for x with his questions to at most d distinct values if and only if a $(k, d+1)$-tree exists, that is, if $d \gtrapprox \frac{2^{k+1}}{ek}(1 + o(1))$.

Tenure Game (J. Spencer)

Two players: the (good) chairman of the department, and the (vicious) dean of the school
The pieces: d non-tenured faculty of the department each at one of k pre-tenured rungs
Winner: The chairman if a faculty is promoted to tenure, otherwise the dean. (A non-tenured faculty becomes tenured if she has rung k and is promoted.)
Procedure: Once each year, the chairman proposes to the dean a subset S of the non-tenured faculty to be promoted by one rung. The dean has two choices: either he accepts the suggestion of the chairman, promotes everybody in S by one rung and fires everybody else, or he does the complete opposite: fires everybody in S and promotes everybody else by one rung.

If all d faculties are at rung 1, then chairman wins iff $k \leq \lfloor \log d \rfloor$.

European Tenure Game (B. Doerr)

Modified Rules: the non-promoted part of the non-tenured faculty is not fired, rather demoted back to rung 1. Assume that all non-tenured faculty are at the lowest rung in the beginning For fixed d let v_d stand for the largest number k of rungs such that the chairman wins.

Doerr (2004) showed

$$\lfloor \log d + \log\log d + o(1) \rfloor \leq v_d \leq \lfloor \log d + \log\log d + 1.73 + o(1) \rfloor.$$

Theorem

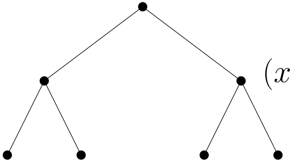
(Gebauer-Sz.-Tardos) The chairman wins the European Tenure Game with d faculty and k rungs if and only if there exists a (k, d)-tree. In particular,

$$v_d = \lfloor \log d + \log\log d + \log e - 1 + o(1) \rfloor.$$

The construction

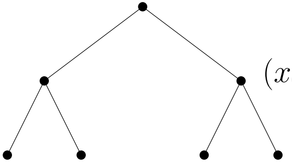
How to construct (k,d)-trees?

Leaf-vectors



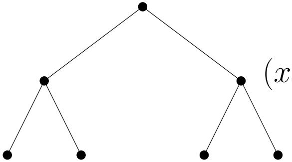
$(x_0, x_1, \ldots, x_i, \ldots, x_k)$

Leaf-vectors



$$(x_0, x_1, \ldots, x_i, \ldots, x_k)$$

$\#$ leaf-descendants at dist. i

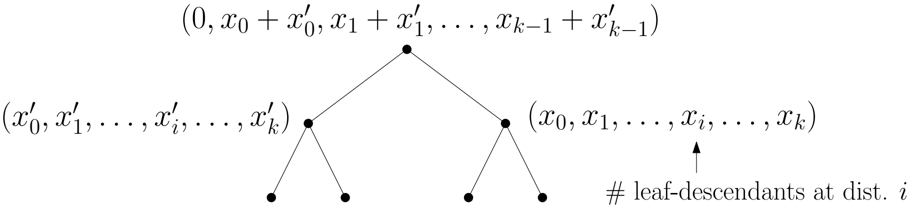


$(x_0, x_1, \ldots, x_i, \ldots, x_k)$

$\#$ leaf-descendants at dist. i

$\sum x_i \leq d$

Leaf-vectors

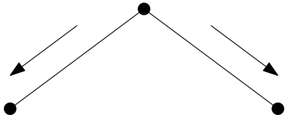


Parents and children

Building the tree from top to bottom by "distributing the debt"

The Fair — SPLIT

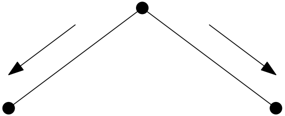
$$(x_0, x_1, \ldots, x_k)$$



Parents and children

Building the tree from top to bottom by "distributing the debt"
The Fair — SPLIT

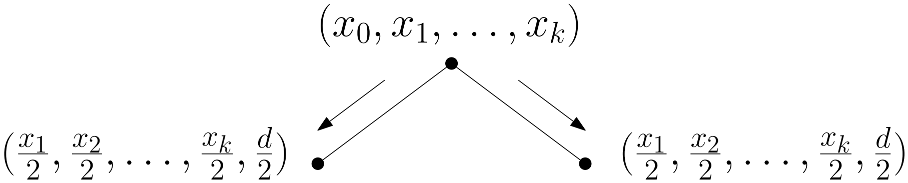
$$(x_0, x_1, \ldots, x_k)$$



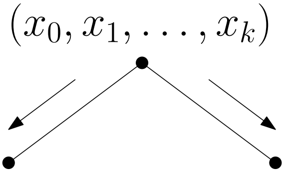
$$\left(\frac{x_1}{2}, \frac{x_2}{2}, \ldots, \frac{x_k}{2}, \frac{d}{2}\right) \qquad \left(\frac{x_1}{2}, \frac{x_2}{2}, \ldots, \frac{x_k}{2}, \frac{d}{2}\right)$$

Building the tree from top to bottom by "distributing the debt"

The Fair — SPLIT



The Unfair — CUT



Parents and children

Building the tree from top to bottom by "distributing the debt"

The Fair — SPLIT

$$(x_0, x_1, \dots, x_k)$$

$$\left(\frac{x_1}{2}, \frac{x_2}{2}, \dots, \frac{x_k}{2}, \frac{d}{2}\right) \qquad \left(\frac{x_1}{2}, \frac{x_2}{2}, \dots, \frac{x_k}{2}, \frac{d}{2}\right)$$

The Unfair — CUT

$$(x_0, x_1, \dots, x_k)$$

$$(x_1, \dots, x_r, 0, \dots, 0) \qquad (0, \dots, 0, x_{r+1}, \dots, x_k, 0)$$

(k,d)-constructible leaf-vectors

For the leaf-vector $\vec{\ell_w} = (x_0, x_1, \ldots, x_k)$ of any vertex w of a (k,d)-tree we have $|\vec{\ell_w}| := \sum x_i \leq d$

(k,d)-constructible leaf-vectors

For the leaf-vector $\vec{\ell_w} = (x_0, x_1, \ldots, x_k)$ of any vertex w of a (k,d)-tree we have $|\vec{\ell_w}| := \sum x_i \leq d$

Def. For a vector $\vec{x}$ with $|\vec{x}| \leq d$ we say that a tree T with root r is a $(k,d,\vec{x})$-tree if

- $\vec{\ell_r} \leq \vec{x}$ (coordinatewise)
- every vertex has $\leq d$ leaf-descendants of distance $\leq k$

For the leaf-vector $\vec{\ell_w} = (x_0, x_1, \ldots, x_k)$ of any vertex w of a (k,d)-tree we have $|\vec{\ell_w}| := \sum x_i \leq d$

Def. For a vector $\vec{x}$ with $|\vec{x}| \leq d$ we say that a tree T with root r is a $(k,d,\vec{x})$-tree if

- $\vec{\ell_r} \leq \vec{x}$ (coordinatewise)
- every vertex has $\leq d$ leaf-descendants of distance $\leq k$

Def. A vector $\vec{x}$ is (k,d)-constructible if there is a $(k,d,\vec{x})$-tree

(k, d)-constructible leaf-vectors

For the leaf-vector $\vec{\ell_w} = (x_0, x_1, \ldots, x_k)$ of any vertex w of a (k, d)-tree we have $|\vec{\ell_w}| := \sum x_i \leq d$

Def. For a vector $\vec{x}$ with $|\vec{x}| \leq d$ we say that a tree T with root r is a $(k, d, \vec{x})$-tree if

- $\vec{\ell_r} \leq \vec{x}$ (coordinatewise)
- every vertex has $\leq d$ leaf-descendants of distance $\leq k$

Def. A vector $\vec{x}$ is (k, d)-constructible if there is a $(k, d, \vec{x})$-tree

Observation

There is a (k, d)-tree $\Leftrightarrow$ $(0, 0, \ldots, 0)$ *is (k, d)-constructible*

Payoff Lemma

Some vectors that are $(k, 2^i)$-constructible:

Payoff Lemma

Some vectors that are $(k, 2^i)$-constructible:

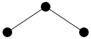
$(1, 0, \ldots, 0)$

Payoff Lemma

Some vectors that are $(k, 2^i)$-constructible:

$(1, 0, \ldots, 0)$

$(0, 2, 0 \ldots, 0)$

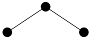


Payoff Lemma

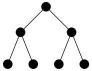
Some vectors that are $(k, 2^i)$-constructible:

$(1, 0, \ldots, 0)$

$(0, 2, 0 \ldots, 0)$

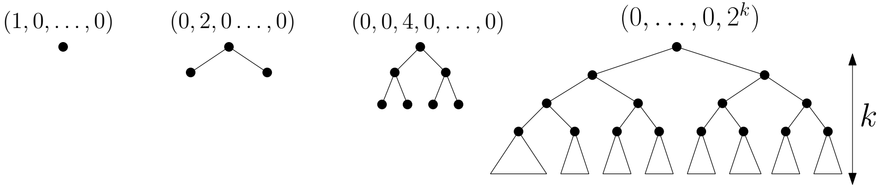


$(0, 0, 4, 0, \ldots, 0)$



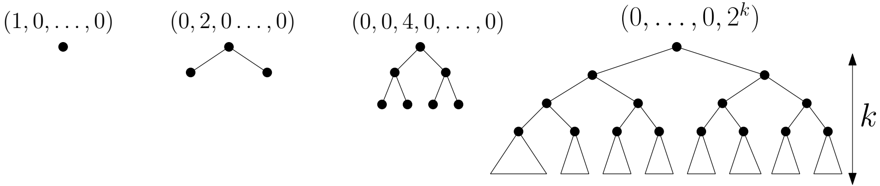
Payoff Lemma

Some vectors that are $(k, 2^i)$-constructible:



Payoff Lemma

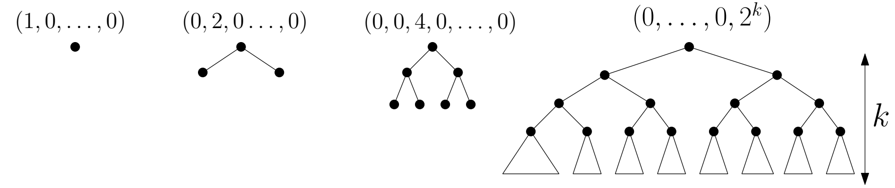
Some vectors that are $(k, 2^i)$-constructible:



Def. weight of $\vec{x}$ $:= w(\vec{x}) := \sum_{i=0}^{k} \frac{x_i}{2^i}$

Payoff Lemma

Some vectors that are $(k, 2^i)$-constructible:



Def. weight of $\vec{x}$ $:= w(\vec{x}) := \displaystyle\sum_{i=0}^{k} \frac{x_i}{2^i}$

Lemma (Payoff Lemma)

Let $|\vec{x}| \leq d$. If $w(\vec{x}) \geq 1$, then $\vec{x}$ is (k, d)-constructible.

Inverse of Kraft's Inequality

The Gebauer-trees

For simplicity assume that $d = 2^{s+1}$ is a power of 2.
Then $\left(0, \ldots, 0, 1, 2, 4, \ldots, \frac{d}{4}, \frac{d}{2}\right)$ is (k, d)-constructible if

$$\sum_{i=1}^{s+1} \frac{d/2^i}{2^{k+1-i}} = (s+1)\frac{d}{2^{k+1}} \geq 1$$

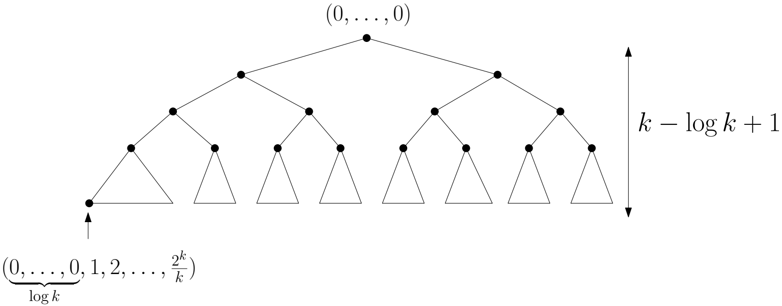
That is, when $d \log_2 d \geq 2^{k+1}$. Holds for $d \approx \frac{2^{k+1}}{k}$.

The Gebauer-trees

For simplicity assume that $d = 2^{s+1}$ is a power of 2.
Then $\left(0,\ldots,0,1,2,4,\ldots,\frac{d}{4},\frac{d}{2}\right)$ is (k,d)-constructible if

$$\sum_{i=1}^{s+1} \frac{d/2^i}{2^{k+1-i}} = (s+1)\frac{d}{2^{k+1}} \geq 1$$

That is, when $d \log_2 d \geq 2^{k+1}$. Holds for $d \approx \frac{2^{k+1}}{k}$.

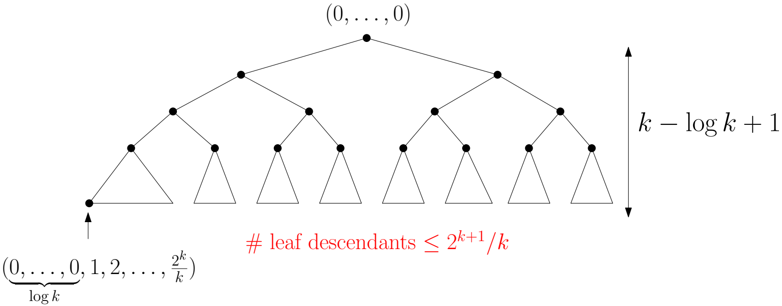


The Gebauer-trees

For simplicity assume that $d = 2^{s+1}$ is a power of 2.
Then $\left(0,\ldots,0,1,2,4,\ldots,\frac{d}{4},\frac{d}{2}\right)$ is (k,d)-constructible if

$$\sum_{i=1}^{s+1} \frac{d/2^i}{2^{k+1-i}} = (s+1)\frac{d}{2^{k+1}} \geq 1$$

That is, when $d\log_2 d \geq 2^{k+1}$. Holds for $d \approx \frac{2^{k+1}}{k}$.



First idea: Cut-and-Split, Left Child pays off

How else can we prove constructibility of vector $(x_0, x_1, \ldots, x_r, x_{r+1}, \ldots, x_k)$?

First idea: Cut-and-Split, Left Child pays off

How else can we prove constructibility of vector $(x_0, x_1, \dots, x_r, x_{r+1}, \dots, x_k)$?

Cut at SMALLEST coordinate r with $\sum_{i=0}^{r-1} \frac{x_{i+1}}{2^i} \geq 1$; just enough so Left Child $(x_1, x_2, \dots, x_r, 0, \dots, 0)$ can immediately pay off

First idea: Cut-and-Split, Left Child pays off

How else can we prove constructibility of vector $(x_0, x_1, \dots, x_r, x_{r+1}, \dots, x_k)$?

Cut at SMALLEST coordinate r with $\sum_{i=0}^{r-1} \frac{x_{i+1}}{2^i} \geq 1$; just enough so Left Child $(x_1, x_2, \dots, x_r, 0, \dots, 0)$ can immediately pay off

Split Right Child $(0, \dots, 0, x_{r+1}, \dots, x_k, 0)$ $\log_2 x_{r+1} =: m$-times and HOPE that with $\left(0, \dots, 0, \frac{x_{r+1}}{2^m}, \frac{x_{r+2}}{2^m}, \dots, \frac{x_k}{2^m}, \frac{d}{2^m}, \frac{d}{2^{m-1}}, \dots, \frac{d}{2}\right)$ the situation is BETTER than with the parent.

First idea: Cut-and-Split, Left Child pays off

How else can we prove constructibility of vector $(x_0, x_1, \dots, x_r, x_{r+1}, \dots, x_k)$?

Cut at SMALLEST coordinate r with $\sum_{i=0}^{r-1} \frac{x_{i+1}}{2^i} \geq 1$; just enough so Left Child $(x_1, x_2, \dots, x_r, 0, \dots, 0)$ can immediately pay off

Split Right Child $(0, \dots, 0, x_{r+1}, \dots, x_k, 0)$ $\log_2 x_{r+1} =: m$-times and HOPE that with $\left(0, \dots, 0, \frac{x_{r+1}}{2^m}, \frac{x_{r+2}}{2^m}, \dots, \frac{x_k}{2^m}, \frac{d}{2^m}, \frac{d}{2^{m-1}}, \dots, \frac{d}{2}\right)$ the situation is BETTER than with the parent.

Repeat this Operation "Cut-and-Split then Left Child pays off"

How else can we prove constructibility of vector $(x_0, x_1, \dots, x_r, x_{r+1}, \dots, x_k)$?

Cut at SMALLEST coordinate r with $\sum_{i=0}^{r-1} \frac{x_{i+1}}{2^i} \geq 1$; just enough so Left Child $(x_1, x_2, \dots, x_r, 0, \dots, 0)$ can immediately pay off

Split Right Child $(0, \dots, 0, x_{r+1}, \dots, x_k, 0)$ $\log_2 x_{r+1} =: m$-times and HOPE that with $\left(0, \dots, 0, \frac{x_{r+1}}{2^m}, \frac{x_{r+2}}{2^m}, \dots, \frac{x_k}{2^m}, \frac{d}{2^m}, \frac{d}{2^{m-1}}, \dots, \frac{d}{2}\right)$ the situation is BETTER than with the parent.

Repeat this Operation "Cut-and-Split then Left Child pays off"

Question Will the sequence of Right Child vectors ever converge to one that can pay off?
How to analyse?

Normalized analytic setting

Set $d = \frac{2}{T} \cdot \frac{2^k}{k}$. Eventually we want to get to $T = e - \epsilon$.

Normalized analytic setting

Set $d = \frac{2}{T} \cdot \frac{2^k}{k}$. Eventually we want to get to $T = e - \epsilon$.

leaf-vector

$\vec{x} = \left(0, \ldots, 0, 1, \ldots, \frac{d}{4}, \frac{d}{2}\right)$

Payoff: $w(\vec{x}) \geq 1$

Normalized analytic setting

Set $d = \frac{2}{T} \cdot \frac{2^k}{k}$. Eventually we want to get to $T = e - \epsilon$.

<u>leaf-vector</u> <u>normalized leaf-vector</u>

$$\vec{x} = \left(0, \ldots, 0, 1, \ldots, \tfrac{d}{4}, \tfrac{d}{2}\right) \rightsquigarrow \vec{y} = (0, \ldots, 0, 1, \ldots, 1, 1)$$

$$\rightsquigarrow y_i = x_i \frac{2^{k+1-i}}{d}$$

Payoff: $w(\vec{x}) \geq 1$

Normalized analytic setting

Set $d = \frac{2}{T} \cdot \frac{2^k}{k}$. Eventually we want to get to $T = e - \epsilon$.

<u>leaf-vector</u>		<u>normalized leaf-vector</u>		<u>leaf-function</u>
$\vec{x} = (0, \ldots, 0, 1, \ldots, \frac{d}{4}, \frac{d}{2})$	$\rightsquigarrow$	$\vec{y} = (0, \ldots, 0, 1, \ldots, 1, 1)$	$\rightsquigarrow$	$f \equiv 1$
	$\rightsquigarrow$	$y_i = x_i \frac{2^{k+1-i}}{d}$		$f : [0,1] \to \mathbb{R}$
Payoff: $w(\vec{x}) \geq 1$				$\int_0^1 f(x)dx \geq T$

For the leaf-function ignore $o(k)$ long segments of the normalized leaf-vector.
(Like the $\Theta(\log k)$ long segment of 0 at the beginning.)

Analytic Cut-and-Split

Let $v \in (0, 1)$.
Operation `Cut-at-`v`-and-Split`
Input function $f : [0, 1] \to \mathbb{R}$

Let $v \in (0,1)$.
Operation `Cut-at-`v`-and-Split`
Input function $f : [0,1] \to \mathbb{R}$
Output

Left Child

$$f_{left}(x) = \begin{cases} 2f(x) & x \in [0,v) \\ 0 & x \in [v,1] \end{cases}$$

Right Child

$$f_{right}(x) = \begin{cases} 2f(x+v) & x \in [0,1-v) \\ 1 & x \in [1-v,1] \end{cases}$$

Analytic Cut-and-Split

Let $v \in (0,1)$.
Operation `Cut-at-`v`-and-Split`
Input function $f : [0,1] \to \mathbb{R}$
Output

Left Child

$$f_{left}(x) = \begin{cases} 2f(x) & x \in [0,v) \\ 0 & x \in [v,1] \end{cases}$$

$\downarrow$

should pay off

$$2\int_0^v f \geq T$$

Right Child

$$f_{right}(x) = \begin{cases} 2f(x+v) & x \in [0,1-v) \\ 1 & x \in [1-v,1] \end{cases}$$

$\downarrow$

should be "better" than parent

does **not** mean "greater integral"

Where to cut?

We perform a series of Cut-and-Splits, cutting at
$1-\delta, 1-2\delta, \dots, 1-N\delta$
for some CONSTANTS $\delta > 0$ and integer N.

Where to cut?

We perform a series of Cut-and-Splits, cutting at
$1-\delta, 1-2\delta, \ldots, 1-N\delta$
for some CONSTANTS $\delta > 0$ and integer N.

At the end of the process the integral of Right Child grows above $T = 2 - \epsilon$ and hence the process stops.

Where to cut?

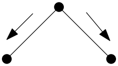
We perform a series of Cut-and-Splits, cutting at
$1-\delta, 1-2\delta, \ldots, 1-N\delta$
for some CONSTANTS $\delta > 0$ and integer N.

At the end of the process the integral of Right Child grows above $T = 2-\epsilon$ and hence the process stops.

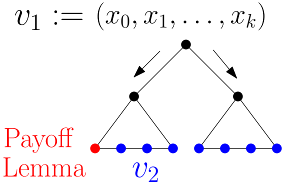
$T = 2-\epsilon$ is the limit of the simple Cut-and-Split.

How to get to $T = e - \epsilon$? — r-deep cuts

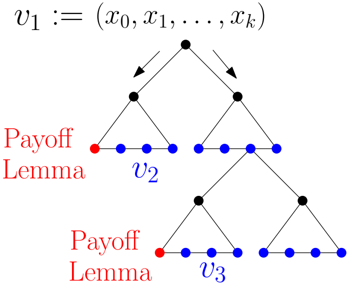
$$v_1 := (x_0, x_1, \ldots, x_k)$$



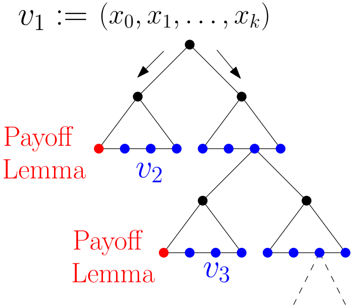
How to get to $T = e - \epsilon$? — r-deep cuts



How to get to $T = e - \epsilon$? — r-deep cuts

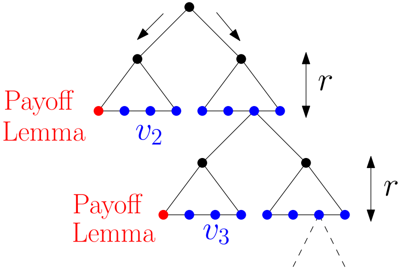


How to get to $T = e - \epsilon$? — r-deep cuts

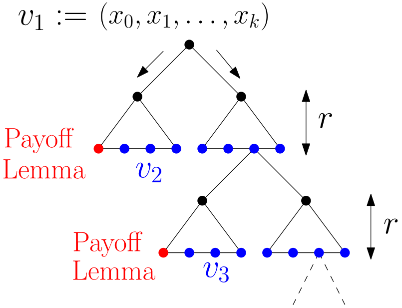


How to get to $T = e - \epsilon$? — r-deep cuts

$v_1 := (x_0, x_1, \ldots, x_k)$

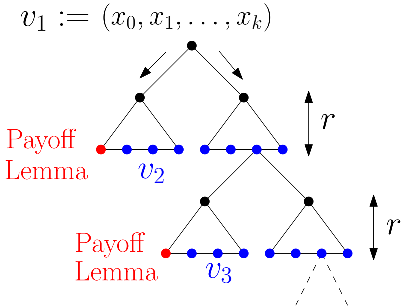


How to get to $T = e - \epsilon$? — r-deep cuts



$$f_{left}(x) = \begin{cases} 2^r f(x) & x \in [0, v) \\ 0 & x \in [v, 1] \end{cases} \qquad f_{right}(x) = \begin{cases} \frac{2^r}{2^r - 1} f(x+v) & x \in [0, 1-v) \\ 1 & x \in [1-v, 1] \end{cases}$$

How to get to $T = e - \epsilon$? — r-deep cuts



$$f_{left}(x) = \begin{cases} 2^r f(x) & x \in [0, v) \\ 0 & x \in [v, 1] \end{cases} \qquad f_{right}(x) = \begin{cases} \frac{2^r}{2^r - 1} f(x+v) & x \in [0, 1-v) \\ 1 & x \in [1-v, 1] \end{cases}$$

$\downarrow$ $\qquad\qquad$ $\downarrow$

Payoff: $2^r \int_0^v f \geq T$ $\qquad$ should be better than parent

How to analyse?

Look at Right Child $f_{right}(x)$ after "time" $t \quad \rightsquigarrow \quad F(t, x)$
(after t/δ infinitesimally small cuts of length δ)

$F(0, x) = 1$ for all $x \in [0, 1]$

$F(t, 1) = 1$ for all $t \geq 0$

How to analyse?

Look at Right Child $f_{right}(x)$ after "time" $t \quad \rightsquigarrow \quad F(t,x)$
(after t/δ infinitesimally small cuts of length δ)

$F(0,x) = 1$ for all $x \in [0,1]$

$F(t,1) = 1$ for all $t \geq 0$

WANT to pay off:

$$\int_0^1 F_{left}(t,x)dx \approx 2^r \delta F(t,0) \geq T$$

So let $r \approx \log_2 \frac{T}{\delta F(t,0)}$

How to analyse?

Look at Right Child $f_{right}(x)$ after "time" $t \quad \rightsquigarrow \quad F(t,x)$
(after t/δ infinitesimally small cuts of length δ)

$F(0,x) = 1$ for all $x \in [0,1]$

$F(t,1) = 1$ for all $t \geq 0$

WANT to pay off:

$$\int_0^1 F_{left}(t,x)dx \approx 2^r \delta F(t,0) \geq T$$

So let $r \approx \log_2 \frac{T}{\delta F(t,0)}$
THEN:

$$F_{right}(t,x) \approx F(t,x) \cdot \frac{2^r}{2^r - 1} \approx F(t,x)\left(1 + \frac{\delta F(t,0)}{T}\right)$$

That is $F(t+\delta, x-\delta) \approx F(t,x)\left(1 + \frac{\delta F(t,0)}{T}\right)$

A differential equation

From the previous page: $F(t+\delta, x-\delta) \approx F(t,x)\left(1+\frac{\delta F(t,0)}{T}\right)$
For some time s, introduce $F_s(t) := F(t, s-t)$.
Rewritten, for any $s-1 \leq t \leq s$:

A differential equation

From the previous page: $F(t+\delta, x-\delta) \approx F(t,x)\left(1+\frac{\delta F(t,0)}{T}\right)$

For some time s, introduce $F_s(t) := F(t, s-t)$.

Rewritten, for any $s-1 \leq t \leq s$:

$$F_s(t+\delta) \quad \approx \quad F_s(t)\left(1+\frac{\delta F(t,0)}{T}\right)$$

A differential equation

From the previous page: $F(t+\delta, x-\delta) \approx F(t,x)\left(1+\frac{\delta F(t,0)}{T}\right)$

For some time s, introduce $F_s(t) := F(t, s-t)$.

Rewritten, for any $s-1 \le t \le s$:

$$\begin{aligned} F_s(t+\delta) &\approx F_s(t)\left(1+\frac{\delta F(t,0)}{T}\right) \\ F_s'(t) \approx \frac{F_s(t+\delta)-F_s(t)}{\delta} &\approx F_s(t)\frac{F(t,0)}{T} \end{aligned}$$

A differential equation

From the previous page: $F(t+\delta, x-\delta) \approx F(t,x)\left(1+\frac{\delta F(t,0)}{T}\right)$

For some time s, introduce $F_s(t) := F(t, s-t)$.

Rewritten, for any $s-1 \le t \le s$:

$$\begin{aligned}
F_s(t+\delta) &\approx F_s(t)\left(1+\frac{\delta F(t,0)}{T}\right) \\
F_s'(t) \approx \frac{F_s(t+\delta)-F_s(t)}{\delta} &\approx F_s(t)\frac{F(t,0)}{T} \\
\int_{s-1}^{s} \frac{F_s'(t)}{F_s(t)}dt &\approx \frac{1}{T}\int_{s-1}^{s} F(t,0)dt
\end{aligned}$$

A differential equation

From the previous page: $F(t+\delta, x-\delta) \approx F(t,x)\left(1+\frac{\delta F(t,0)}{T}\right)$
For some time s, introduce $F_s(t) := F(t, s-t)$.
Rewritten, for any $s-1 \le t \le s$:

$$
\begin{aligned}
F_s(t+\delta) &\approx F_s(t)\left(1+\frac{\delta F(t,0)}{T}\right)\\
F_s'(t) \approx \frac{F_s(t+\delta)-F_s(t)}{\delta} &\approx F_s(t)\frac{F(t,0)}{T}\\
\int_{s-1}^{s}\frac{F_s'(t)}{F_s(t)}dt &\approx \frac{1}{T}\int_{s-1}^{s}F(t,0)dt\\
\ln F_s(s) - \ln F_s(s-1) &\gtrapprox \frac{F(s-1,0)}{T}
\end{aligned}
$$

A differential equation

From the previous page: $F(t+\delta, x-\delta) \approx F(t,x)\left(1+\frac{\delta F(t,0)}{T}\right)$
For some time s, introduce $F_s(t) := F(t, s-t)$.
Rewritten, for any $s-1 \leq t \leq s$:

$$
\begin{aligned}
F_s(t+\delta) &\approx F_s(t)\left(1+\frac{\delta F(t,0)}{T}\right)\\
F_s'(t) \approx \frac{F_s(t+\delta)-F_s(t)}{\delta} &\approx F_s(t)\frac{F(t,0)}{T}\\
\int_{s-1}^{s}\frac{F_s'(t)}{F_s(t)}dt &\approx \frac{1}{T}\int_{s-1}^{s}F(t,0)dt\\
\ln F_s(s) - \ln F_s(s-1) &\gtrapprox \frac{F(s-1,0)}{T}\\
\ln F(s,0) &\gtrapprox \frac{F(s-1,0)}{T}
\end{aligned}
$$

A differential equation

From the previous page: $F(t+\delta, x-\delta) \approx F(t,x)\left(1+\frac{\delta F(t,0)}{T}\right)$

For some time s, introduce $F_s(t) := F(t, s-t)$.

Rewritten, for any $s-1 \leq t \leq s$:

$$\begin{aligned}
F_s(t+\delta) &\approx F_s(t)\left(1+\frac{\delta F(t,0)}{T}\right) \\
F_s'(t) \approx \frac{F_s(t+\delta)-F_s(t)}{\delta} &\approx F_s(t)\frac{F(t,0)}{T} \\
\int_{s-1}^{s} \frac{F_s'(t)}{F_s(t)} dt &\approx \frac{1}{T}\int_{s-1}^{s} F(t,0)dt \\
\ln F_s(s) - \ln F_s(s-1) &\gtrapprox \frac{F(s-1,0)}{T} \\
\ln F(s,0) &\gtrapprox \frac{F(s-1,0)}{T}
\end{aligned}$$

If $F(s,0)$ converged to a finite limit a, we would have $T \geq \frac{a}{\ln a} \geq e$.

So $\int_0^1 F(s,x)dx \approx F(s,0) \to \infty$ and the right child pays off. $\square$

Def. Let $D(k)$ be the largest integer such that for every k-uniform hypergraph $(X, \mathcal{F})$ with $\Delta(\mathcal{F}) \le D(k)$ Breaker has a winning strategy.

Def. Let $D(k)$ be the largest integer such that for every k-uniform hypergraph $(X, \mathcal{F})$ with $\Delta(\mathcal{F}) \leq D(k)$ Breaker has a winning strategy.

Neighborhood Conjecture There is an $\epsilon > 0$ such that $D(k) > (1+\epsilon)^k$.

Def. Let $D(k)$ be the largest integer such that for every k-uniform hypergraph $(X, \mathcal{F})$ with $\Delta(\mathcal{F}) \leq D(k)$ Breaker has a winning strategy.

Neighborhood Conjecture There is an $\epsilon > 0$ such that $D(k) > (1+\epsilon)^k$.

More modest goal: $D(k) > 0.51k$

Open problems — The Neighborhood Conjecture

Def. Let $D(k)$ be the largest integer such that for every k-uniform hypergraph $(X, \mathcal{F})$ with $\Delta(\mathcal{F}) \leq D(k)$ Breaker has a winning strategy.
Neighborhood Conjecture There is an $\epsilon > 0$ such that $D(k) > (1+\epsilon)^k$.
More modest goal: $D(k) > 0.51k$
It is still possible that some $\epsilon = \epsilon(k) \to 1$ could be chosen.