Computing Non-Repetitive Sequences Using the Lovász Local Lemma

Daniel Mourad

University of Connecticut

Daniel.Mourad@Uconn.edu

Online Logic Seminar
Southern Illinois University
05/04/2023

Introduction and Background

LLL Background

- The Lovász local lemma (LLL) is a technique from the probabilistic method for showing the existence of *finite objects*.
- The original LLL by Erdös and Lovász (1975) is non-constructive.
- Finite and non-constructive - at first, LLL does not seem useful for computability theory.

Constructive LLL

- Much effort was put into constructivising the LLL.
- Moser and Tardos (2010) made a breakthrough with their resampling algorithm.
- Constructive but still finite - one step closer being computable.

LLL Background

- Rumyantsev and Shen (2014) used the Moser-Tardos algorithm to prove an effective version of the LLL (CLLL).
- CLLL has been used in the reverse math of combinatorial theorems (Liu, Monin, and Patey, 2018; Csima, Dzhafarov, Hirschfeldt, Jockusch, Solomon, and Westrick, 2019; Hirschfeldt and Reitzes, 2022)

Goals

- What else can CLLL be used for?
- Can generalizations of the LLL also be effectivised?

We will effectivise combinatorial theorems proven using LLL but which fall outside of the scope of CLLL. To do this, we will use a "lefthanded" version of the CLLL based on a "lefthanded" version of the LLL.

Lovász Local Lemma

$$\Pr(A) \leq z(A) \prod_{B \in \Gamma(A)} (1 - z(B)).$$

Figure: The main condition of the LLL (not important for this talk!).

The Probabilistic Method

The **Probabilistic Method** is a technique from combinatorics for showing existence results.

- Suppose you want to show that there is an $X \in \mathcal{F}$ with the property P. Then,
  1. Specify a way to "randomly choose" a $Y \in \mathcal{F}$.
  2. Show that the $\Pr(Y \text{ has property } P) > 0$.
  3. Therefore, there is a $X \in \mathcal{F}$ with property P.

Example: If $\mathcal{F}$ is the class of binary strings of size n, we can "randomly choose" a $Y \in \mathcal{F}$ by flipping a coin for each entry of the string.

Mutually Independent Events

Suppose that $\mathcal{A} = \{A_1, A_2, \ldots A_r\}$ is a finite set of "bad" events that we wish to avoid.

If the bad events are **mutually independent** and each has positive probability to be false, then they have positive probability to *all* be false.

$$\Pr\left(\bigcap_{A \in \mathcal{A}} \overline{A}\right) = \prod_{A \in \mathcal{A}} \Pr(\overline{A}) > 0$$

In particular, $\Pr\left(\bigcap_{A \in \mathcal{A}} (\bar{A})\right) > 0$ implies that it is *nonempty* as a *set*.

The LLL generalizes this by allowing dependence within $\mathcal{A}$ in exchange for stricter requirements on their probabilities.

Mutually Independent Events

Suppose that $\mathcal{A} = \{A_1, A_2, \ldots A_r\}$ is a finite set of "bad" events that we wish to avoid.

If the bad events are **mutually independent** and each has positive probability to be false, then they have positive probability to *all* be false.

$$\Pr\left(\bigcap_{A \in \mathcal{A}} \overline{A}\right) = \prod_{A \in \mathcal{A}} \Pr(\overline{A}) > 0$$

In particular, $\Pr\left(\bigcap_{A \in \mathcal{A}} (\bar{A})\right) > 0$ implies that it is *nonempty* as a *set*.

The LLL generalizes this by allowing dependence within $\mathcal{A}$ in exchange for stricter requirements on their probabilities.

$$\Pr(A) \leq z(A) \prod_{B \in \Gamma(A)} (1 - z(B)).$$

The Lovász Local Lemma

Lemma (Erdős and Lovász, 1975)

Let $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ *be events in an arbitrary probability space. Let* $\Gamma(A_i) \subset \mathcal{A}$ *be such that* A_i *is mutually independent of all* $A_j \notin \Gamma(A_i)$*.*
If the probabilities of each $A_i \in \mathcal{A}$ *are small enough relative to the size* $\Gamma(A_i)$*, then*

$$\Pr\left(\bigcap_{A \in \mathcal{A}} \overline{A}\right) > 0$$

Remark: To be precise, "small enough relative to the sizes of the $\Gamma(A)$", means that there exists a function $z : \mathcal{A} \to \mathbb{R}$ such that for each $A_i \in \mathcal{A}$,

$$\Pr(A_i) \leq z(A_i) \prod_{B \in \Gamma(A_i)} (1 - z(B)).$$

Satisfiability of Sentences in Conjunctive Normal Form

A crucial application of the LLL is to satisfiability of propositional formulas in conjunctive normal form.

Fix propositional variables $x_1, x_2, x_3, \ldots$ with values ranging over $\{T, F\}$.

Definition (Conjunctive Normal Form)

A formula is in conjunctive normal form (CNF) if it is a (possibly countable) conjunction of disjunctive clauses over the x_i and $\neg x_i$.

Example

$\mathcal{C}_1 \equiv (x_3 \vee \neg x_7 \vee x_9 \vee x_{10}) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_9)$ is a CNF with three clauses.

The **length** of a clause is the number of x_i in the clause. If two clauses share a variable, then they are called **neighbors**. A CNF is considered **satisfiable** if there is a valuation of the x_i which makes it true.

CNF Examples

Example

$\mathcal{C}_2 \equiv (x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$ is a CNF that is not satisfiable.

Example

$\mathcal{C}_3 \equiv \bigwedge_{n=1}^{\infty}(x_{2n} \vee x_{2n+1} \vee x_{2n+2}) = (x_2 \vee x_3 \vee x_4) \wedge (x_4 \vee x_5 \vee x_6) \wedge \ldots$ is a CNF with countably many clauses, each with length 3 and, except for the first clause, exactly 2 neighbors each.

Example

The compactness theorem of propositional logic states that any infinite non-satisfiable CNF has a non-satisfiable finite initial segment.

CNFs and LLL

Example

Let $m > 3$. A finite CNF such that each clause is of length at least m and has at most 2^{m-2} many neighbors is satisfiable.

Proof (sketch).

Let $\mathcal{F}$ be the set of valuations of the propositional variables.

- Let $\Pr(x_i) = 1/2$ for each propositional variable x_i.
- Let $A_i = \{\text{the } i\text{'th clause is false}\}$. Then, $\Pr(A_i) \leq \frac{1}{2^m}$.
- Let $\Gamma(A_i) = \{A_j : \text{the } i\text{'th and } j\text{'th clauses are neighbors}\}$.
- To use the LLL, show that each $\Pr(A_i)$ is "small enough" compared to the size (and probabilities) of its neighborhood.

□

The Resample Algorithm: Constructive and Computable Lovász Local Lemma

Figure: Bad events are monochromatic triangles.

Local Lemma Setup (Variable Context)

Algorithmic versions of the LLL benefit from a *variable context*, although there are generalizations beyond it (Harvey and Vondrak, 2020).

The variable context consists of

- The set $\mathcal{A} = \{A_1, A_2, \ldots A_r\}$ of "bad" events.
  - CNF: A_i is the event that the i'th clause is false.
- A set $\mathcal{X} = \{x_1, x_2, \ldots x_{M-1}\}$ of mutually independent discrete random variables with finite ranges.
  - CNF: x_i is the i'th propositional variable with $\Pr(x_i = T) = \Pr(x_i = F) = 1/2$
- For each $A_i \in \mathcal{A}$ a variable set $\mathbf{vbl}(A_i) \subset \mathcal{X}$ such that truth of A_i is completely determined by the valuation of the variables in $\mathbf{vbl}(A_i)$.
  - CNF: $\mathbf{vbl}(A_i)$ is the set of variables present in the i'th clause.
  - $A_i \in \Gamma(A_j)$ if the i'th clause and the j'th clause share a variable.

Moser-Tardos Algorithm

- The resample algorithm tries to find a valuation of $\mathcal{X} = \{x_0, x_1, \ldots, x_{M-1}\}$ that makes each $A \in \mathcal{A}$ false.
- When $A \in \mathcal{A}$ is true, the resample algorithm takes fresh new samples of $\mathsf{vbl}(A) \subset \mathcal{X}$ and keeps the other variables the same.
- If multiple bad events are true, the decision of which gets resampled first does not matter.
- Once all of the bad events are false, the resample algorithm halts.

Example Stage

Suppose we are applying the resample algorithm to

$$\mathcal{C} = (\neg x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_1 \vee x_3).$$

Let A_i be the event that i'th clause is **false**. We say that A_i is **good** if the i'th clause is **true** (so A_i is good if it is false).

Example

- Suppose the initial valuation is $(x_1, x_2, x_3) = (F, F, T)$.
  - Then, A_1 is good, A_2 is bad, and A_3 is good.
- We resample one of the bad events: we can only pick A_2.
  - $\mathrm{vbl}(A_2) = \{x_2, x_3\}$, so we flip new coins for x_2 and x_3.
- Suppose the result is $(x_1, x_2, x_3) = (F, F, F)$.
  - Now, A_2 is now good!
  - However, A_3 became bad. In the next stage, we will resample it.

Runtime

Theorem (Moser and Tardos, 2010)

If the set of events $\mathcal{A}$ satisfies the hypothesis of the LLL, then, with probability 1, the resample algorithm halts on an assignment of the $x_0, x_1, \ldots, x_{M-1}$ that makes each $A \in \mathcal{A}$ false.
Furthermore, the expected value of the halting time τ for the resample algorithm is "basically" linear in the size of $\mathcal{A}$.

By "basically" linear, we mean that

$$\mathbb{E}(\tau) \leq \sum_{A \in \mathcal{A}} \frac{z(A)}{1 - z(A)}$$

so it *is* linear if there is a maximum value on $\Pr(A)$.

Computable LLL

Infinite LLL

Theorem (Beck, 1984)

Let $\mathcal{A}$, $\mathcal{X} = \{x_0, x_1, \dots\}$ form a variable context such that he variable set $\mathrm{vbl}(A)$ is finite for each $A \in \mathcal{A}$.
Then, if each finite subset of $\mathcal{A}$ satisfies the conditions of the LLL, we have that there exists a valuation of $\mathcal{X}$ making each $A \in \mathcal{A}$ false.

- By the LLL, for each M, there is a valuation of $\mathcal{X}_M = \{x_0, x_1, \dots, x_{M-1}\}$ that makes each relevant $A \in \mathcal{A}$ false (relevant meaning that $\mathrm{vbl}(A) \subset \mathcal{X}_M$).
- These finite valuations form an infinite finitely branching tree.
- By Königs lemma, this tree has a path. This path is the required valuation. By low basis theorem (Jockusch and Soare, 1972), there is a path with low Turing degree.

Question

Is there an algorithm which effectivizes this use of König's Lemma?

Computable Local Lemma Setup

- Let $\mathcal{X} = \{x_1, x_2, \dots\}$ be a **countable** set of random variables with finite ranges.
- Let $\mathcal{A} = \{A_1, A_2, \dots\}$ be a **countable** set of events such that
  - Each $A \in \mathcal{A}$ is determined by a finite set of variables $\mathrm{vbl}(A) \subset \mathcal{X}$.
  - ⋆ For each $A \in \mathcal{A}$, the set of neighbors $\Gamma(A) = \{B \in \mathcal{A} : \mathrm{vbl}(A) \cap \mathrm{vbl}(B) \neq \emptyset\}$ is finite.
  - ⋆ For each x_i, $\{A_j : x_i \in \mathrm{vbl}(A_j)\}$ is finite.

Along with some reasonable computability conditions (can you think of what is required?).

Computable Lovász Local Lemma

Theorem (Computable Local Lemma (CLL) Rumyantsev and Shen, 2014)

Under the previously stated conditions, if the neighborhoods of events in $\mathcal{A}$ is small enough compared to their probabilities, then there is a **computable** *assignment of the $x_0, x_1 \ldots$ which makes each $A_i \in \mathcal{A}$ false.*

Proof Sketch: Computable Lovász Local Lemma

First, we note that the infinitary resample algorithm behaves analogously to the finite case.

Lemma (Rumyantsev and Shen, 2014)

Given the conditions of the CLLL,

$$\Pr(\textit{All } x_i \textit{ are resampled finitely many times each}) = 1.$$

Furthermore, let y_i be the final value of x_i. Then,

$$\Pr(\textit{The valuation } y_0, y_1, y_2 \ldots \textit{ makes each } A \in \mathcal{A} \textit{ false}) = 1.$$

However, at a fixed stage s, we cannot guarantee that any x_i does not get resampled after stage s, so we cannot compute $y_0, y_1, y_2 \ldots$.

Proof Sketch: Computable Lovász Local Lemma

Lemma (Rumyantsev and Shen, 2014)

There is a computable function $f : \omega^2 \to \mathbb{Q}$ *such that*

$$\Pr(x_i \textit{ is resampled after stage } s) \leq f(i, s)$$

and $\lim_{s \to \infty} f(i, s) = 0$ *for every* i.

To compute initial segment $y_0, \ldots, y_{M-1}$ of a witness to the infinite LLL,

- Simulate the resample algorithm for every possible result of each resampling of each x_i up to the required use.
- Do this for a large enough number of steps s to approximate the probability distribution on the final values $y_0, \ldots, y_{M-1}$.
- Pick a valuation of $y_0, \ldots y_{M-1}$ that has approximate probability greater than $\sum_{i=1}^{M} f(i, s)$.

Applications of CLLL

The CLLL has been used to show that there are computable colorings $c : \omega \to \{0, 1\}$ such that (not simultaneously)

- each $\mathsf{HT}^{=2}$ solution to c is DNC relative to $0'$ (CDHJSW, 2019).
- each thin-$\mathsf{HT}^{=2}$ solution to c is DNC relative to $0'$ (Hirschfeldt and Reitzes, 2022).
- each $\mathsf{OVW}(2, 2)$-solution to c is DNC relative to $0'$ (Liu, Monin, and Patey, 2018), with c interpreted as coloring $c : 2^{<\omega}$ of finite binary strings.

Non-Repetitive Sequences

$$a_0 a_1 a_2 \ldots a_{11} = 011010010001,$$

Figure: An example of repetition in binary sequence.

Beck's Theorem

Jósef Beck (1984) used the infinite LLL to prove the existence of certain types of non-repetitive sequences.

Theorem (M.)

Beck's theorem is computably true.

Proof method:

- CLLL is the natural first choice. However, Beck's theorem does not satisfy the conditions.
- We develop a lefthanded version of the CLLL based on Pegdens (2011) lefthanded version of the LLL.

Repetition

Example

In the string

$$a_0a_1a_2\dots a_{11} = 011010010001,$$

the only pair of identical blocks of size 4 are $[3, 7)$ and $[6, 10)$.

Question

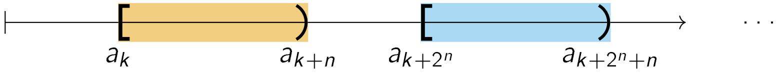
How non-repetitive can an infinite binary sequence be?

Forced Repetition

Given any sequence, there is a hard minimum on how long you need to look before you find two identical blocks of fixed length n.

Proof:

- There are exactly $2^n + 1$ many blocks of length n between $[k, k+n)$ and $[k+2^n, k+2^n+n)$.



- There are 2^n many binary sequences of length n.
- By pigeonhole principle, two of the blocks must be identical.

Hence there will always be a repetition of a block of length n within distance 2^n.

Beck's Theorem

If you only care about long sequences, these repetitions can be forced to be almost as far apart as possible.

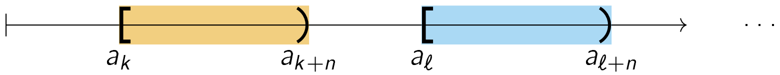
Theorem (Beck, 1984)

For each $\varepsilon > 0$ there is an N_ε and an infinite $\{0,1\}$-valued sequence such that any two identical blocks

$$[k, k+n) \quad \textit{and} \quad [\ell, \ell+n)$$

of length $n > N_\varepsilon$ have distance

$$\ell - k > (2-\varepsilon)^n.$$



Proof Sketch

Theorem (Beck, 1984)

For each $\varepsilon > 0$ there is an N_ε and an infinite $\{0,1\}$-valued sequence such that any two identical blocks $[k, k+n)$ and $[\ell, \ell+n)$ of length $n > N_\varepsilon$ have distance $\ell - k$ greater than $(2-\varepsilon)^n$.

- Let $A_{k,\ell,n} = A_{[k,k+n],[\ell,\ell+n)}$ be the event that $[k, k+n)$ is identical to $[\ell, \ell+n)$.
  - $A_{[k,k+n),[\ell,\ell+m)} \in \mathcal{A}$ if and only if $k < \ell < k + (2-\varepsilon)^n$.
- Let the random variable x_i represent the value of the i'th entry.
  - $\Pr(x_i = 1) = \Pr(x_i = 0) = \frac{1}{2}$ for all i.
- $\mathrm{vbl}(A_{[k,k+n),[\ell,\ell+n)}) = [k, k+n) \cup [\ell, \ell+n)$.

Now apply the infinite LLL. □

To effectivise: try using CLLL.

Unsatisfied Conditions

The conditions of the CLLL do not hold for Beck's theorem!

The CLLL requires that each x_i can be resampled by finitely many of the $A \in \mathcal{A}$.

However, x_i can be resampled infinitely many events.

- Fix k. The $x_k \in \text{vbl}(A_{[k,k+n),[\ell,\ell+n)})$ for all n large enough relative to ℓ.

We do not have probability 1 that each x_i is resampled finitely many times.

Idea: Why bother resampling $[k, k+n)$?

Modified Resample Algorithm: Lefthanded Computable Local Lemma

Modified Resample Algorithm for Non-repetitive Sequences

We can fix this problem by only resampling the rightmost of a pair of identical intervals; If $A_{[k,k+n),[\ell,\ell+n)}$ is true, only resample

$$\mathrm{rsp}(A_{[k,k+n),[\ell,\ell+n)}) := [\ell, \ell + n).$$

01101001000 01

Figure: $\mathrm{rsp}\left(A_{[3,7),[6,10)}\right) = [6, 10)$.

We also leverage the order of resampling.

At each stage, resample $A_{[k_0,k_0+n_0),[\ell_0,\ell_0+n_0)} \in \mathcal{A}$ such that ℓ_0 is minimal among true $A_{[k,k+n),[\ell,\ell+n)} \in \mathcal{A}$.

Example Stage (Lefthanded Resample Algorithm)

Recall that $A_{[k,k+n),[\ell,\ell+n)}$ denotes the "event" that $[k, k+n)$ and $[\ell, \ell+n)$ are identical. $\mathcal{A}$ denotes the set of such events whose intervals are too close together.

Example

Suppose that $A_{[3,7),[10,14)}, A_{[0,4),[5,9)} \in \mathcal{A}$. If the current valuation is

$$a_0, a_1, \ldots, a_{13} = 011\,0100\,101\,\underline{0100},$$

then $A_{[3,7),[10,13)}$ is true. So, the lefthanded resample algorithm takes new random samples for each of the underlined entries. A possible result (with probability 2^{-4}) is

$$a_0, a_1, \ldots, a_{13} = 0110\,1\,0010\,1\underline{0111}.$$

Resampling $A_{[3,7),[10,14)}$ **cannot** cause $A_{[0,4),[5,9)}$ to go from false (good) to true (bad).

Modified Resample Algorithm: General Case

Start with the conditions of the computable LLL.

- For each $A \in \mathcal{A}$, split $\mathsf{vbl}(A)$ into sets $\mathsf{stc}(A)$ and $\mathsf{rsp}(A)$ such that
  - $\max(\mathsf{stc}(A)) < \min(\mathsf{rsp}(A))$.
  - $\mathsf{rsp}(A)$ is an interval.
  - $\mathsf{vbl}(A) \subset (\mathsf{rsp}(A) \cup \mathsf{stc}(A))$.
- Prioritize the left-most true event. If

$$\max(\mathsf{rsp}(A)) < \max(\mathsf{rsp}(B))$$

  then prioritize A before B.
- When A is true and of maximal priority, only resample the $x \in \mathsf{rsp}(A)$ and keep the $x \in \mathsf{stc}(A)$ the same.
- Define $\Gamma_{\mathsf{rsp}}(A) = \{B : \mathsf{rsp}(A) \cap \mathsf{rsp}(B) \neq \emptyset\}$.

Theorem (Lefthanded Computable Lovász Local Lemma (LCLLL))

Suppose all the conditions of the CLLL are true, with exceptions that

- *Need that* x_i *in finitely many* rsp *sets, instead of needing* x_i *in finitely many* vbl *sets.*
- *Replace the LLL condition by*

$$P^*(A) \leq \alpha z(A) \prod_{B \in \Gamma_{\mathrm{rsp}}(A)} (1 - z(B))$$

for some function $P^*(A)$ *such that for each valuation* μ *of the variables in* $\mathrm{stc}(A)$,

$$P^*(A) \geq \Pr(A | x = \mu(x) \text{ for all } x \in \mathrm{stc}(A)).$$

Then, there is a computable assignment of the variables in $\mathcal{X}$ *which makes each event in* $\mathcal{A}$ *false.*

Applications

The LCLLL allows us to effectivise Beck's Theorem.

Theorem (Effective Version of Beck's Theorem)

There are witnesses to Beck's theorem uniformly computable in ε*. That is, there is a computable function* q *such that* $\Phi_{q(n)}$ *computes a non-repetitive sequence for* $\varepsilon = 1/n$*.*

Another Application

The LCLLL can also compute sequences whose adjacent intervals are very different. The following is an effective version of an exercise from *The Probabilistic Method* by Alon, Spencer, and Erdős.

Theorem

There is a computable function q such that $\Phi_{q(n)}$ computes a witness to the following statement for $\varepsilon = 1/n$.
For each $\varepsilon > 0$, there is an N_ε and a computable $\{0, 1\}$-valued sequence such that any two adjacent intervals of length $n > N_\varepsilon$ have share at most $(\frac{1}{2} - \varepsilon)n$ many entries.

Runtime

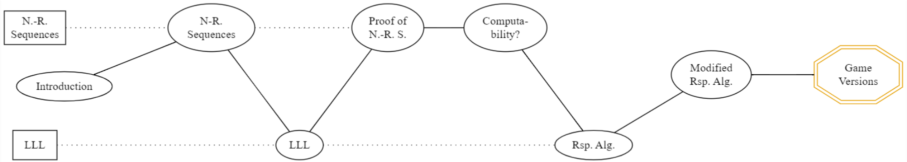
We also get the following purely algorithmic extension of Moser and Tardos' result.

Let τ'_M be the first stage of the modified resample algorithm at which $A_1, \ldots, A_M$ all false.

Theorem

Under the same conditions as the lefthanded computable local lemma, the expected value of τ_M is finite and "basically" linear in M.

Game Versions



Binary Sequence Games

Pegden (2011) studies two player game versions of Beck's theorem.

Binary sequence game:

- Two players take turns selecting bits in a binary sequence.
- The binary sequence game generates the sequence

$$a_1 a_2 a_3 a_4 a_5 a_6 \cdots = e_1 d_2 e_3 d_4 e_5 d_6 \ldots$$

- The e's are chosen by player 1
- The d's are chosen by player 2.
- Both players have knowledge of all previous bits before playing the next one.

Strategies Which Force Non-Repetitiveness

Theorem (Pegden, 2011)

For every $\varepsilon > 0$ *there is an* N_ε *such that Player 1 has a strategy in the binary sequence game ensuring that any two identical blocks of length* $n > N_\varepsilon$ *have distance at least* $f(n) = (2-\varepsilon)^{n/2}$.

To prove this, Pegden develops a version of the LLL called the *lefthanded local lemma*. The mechanism it uses and the problem it solves are similar to LCLLL.

Computably Defeating a Strategy

The LCLLL effectivises Pegden's use of the lefthanded local lemma.

Theorem

For every $\varepsilon > 0$ there is an N_ε such that for each player 2 strategy g in the binary sequence game, there is a g-computable sequence $e_1 e_3 e_5 \ldots$ of player 1 moves that when played against g, any two identical blocks of length $n > N_\varepsilon$ in the resulting sequence have distance at least $f(n) = (2 - \varepsilon)^{n/2}$.

Full Strategies

However, Pegden uses these counter-strategies in a (open) determinacy argument, showing that player 2 cannot have a winning strategy. Therefore, player 1 has a winning strategy.

Effectivising the determinacy argument is outside the scope of the LCLLL.

Further Questions

- Are strategies to Pegden's binary sequence games computable?
  - The determinacy argument can be replaced by a compactness argument, so we at least know that they form a Π^0_1 class.
- Are there more applications of the CLLL and the LCLLL?
- The LCLLL is a "lefthanded version of the computable local lemma." Can we get a "computable version of the lefthanded local lemma?"

Effective Local Lemma Setup

- Let $\mathcal{X} = \{x_1, x_2, \ldots\}$ be a set of random variables with finite ranges and probability distributions.
- Let $\mathcal{A} = \{A_1, A_2, \ldots\}$ be a set of events such that
  - Each $A \in \mathcal{A}$ is determined by a finite set of variables $\text{vbl}(A) \subset \mathcal{X}$.
  - The $\text{vbl}(A_i)$.
  - For each $A \in \mathcal{A}$, the set of neighbors $\Gamma(A) = \{B \in \mathcal{A} : \text{vbl}(A) \cap \text{vbl}(B) \neq \emptyset\}$ is finite.
  - For each x_i, $\{A_j : x_i \in \text{vbl}(A_j)\}$ is finite and .

Effective Local Lemma Setup

- Let $\mathcal{X} = \{x_1, x_2, \ldots\}$ be a set of random variables with finite ranges and uniformly computable probability distributions.
- Let $\mathcal{A} = \{A_1, A_2, \ldots\}$ be a set of events such that
  - Each $A \in \mathcal{A}$ is determined by a finite set of variables $\mathrm{vbl}(A) \subset \mathcal{X}$ and the truth of A is uniformly computable from any valuation of $\mathcal{X}$.
  - The code numbers for $\mathrm{vbl}(A_i)$ are uniformly computable with respect to i.
  - For each $A \in \mathcal{A}$, the set of neighbors $\Gamma(A) = \{B \in \mathcal{A} : \mathrm{vbl}(A) \cap \mathrm{vbl}(B) \neq \emptyset\}$ is finite.
  - For each x_i, $\{A_j : x_i \in \mathrm{vbl}(A_j)\}$ is finite and

.

Effective Local Lemma Setup

- Let $\mathcal{X} = \{x_1, x_2, \ldots\}$ be a set of random variables with finite ranges and uniformly computable probability distributions.
- Let $\mathcal{A} = \{A_1, A_2, \ldots\}$ be a set of events such that
  - Each $A \in \mathcal{A}$ is determined by a finite set of variables $\mathrm{vbl}(A) \subset \mathcal{X}$ and the truth of A is uniformly computable from any valuation of $\mathcal{X}$.
  - The code numbers for $\mathrm{vbl}(A_i)$ are uniformly computable with respect to i.
  - For each $A \in \mathcal{A}$, the set of neighbors $\Gamma(A) = \{B \in \mathcal{A} : \mathrm{vbl}(A) \cap \mathrm{vbl}(B) \neq \emptyset\}$ is finite.
  - For each x_i, $\{A_j : x_i \in \mathrm{vbl}(A_j)\}$ is finite and has code number uniformly computable with respect to i.

Thank you!

References I

Alon, Noga, Joel Spencer, and Paul Erdős (1992). *The Probabilistic Method*. John Wiley & Sons. ISBN: 0-471-53588-5.

Beck, Józef (1984). "an Application of Lovasz Local Lemma: There Exists an Infinite 01-Sequence Containing No Near Identical Intervals". In: *Finite and Infinite Sets* 37, pp. 103–107. DOI: 10.1016/b978-0-444-86893-0.50011-5.

Csima, Barbara, Damir Dzhafarov, Denis Hirschfeldt, Carl Jockusch, Reed Solomon, and Linda Westrick (2019). "The reverse mathematics of Hindman's Theorem for sums of exactly two elements". In: *Computability* 8 (3-4), pp. 253–263. ISSN: 22113576. DOI: 10.3233/COM-180094.

Erdős, P. and L. Lovász (1975). "Problems and results on 3-chromatic hypergraphs and some related questions". In: *Infinite and finite sets* 2 (2), pp. 609–627.

References II

Harvey, Nicholas J.A. and Jan Vondrak (2020). "An algorithmic proof of the lovasz local lemma via resampling oracles". In: *SIAM Journal on Computing* 49 (2), pp. 394–428. ISSN: 10957111. DOI: 10.1137/18M1167176.

Hirschfeldt, Dennis and Sarah Reitzes (2022). "Thin Set Versions of Hindman's Theorem". In: *Submitted*.

Jockusch, Carl G. and Robert I. Soare (1972). "$\prod_1^0$ Classes and Degrees of Theories". In: *Transactions of the American Mathematical Society* 173, pp. 33–56. ISSN: 00029947. (Visited on 01/05/2023).

Liu, Lu, Benoit Monin, and Ludovic Patey (2018). "A computable analysis of variable words theorems". In: *Proceedings of the American Mathematical Society* 147 (2), pp. 823–834. ISSN: 0002-9939. DOI: 10.1090/proc/14269.

Moser, Robin A. and Gábor Tardos (2010). "A constructive proof of the general lovász local lemma". In: *Journal of the ACM* 57 (2), pp. 1–12. ISSN: 00045411. DOI: 10.1145/1667053.1667060.

References III

Pegden, Wesley (2011). "Highly nonrepetitive sequences: Winning strategies from the local lemma". In: *Random Structures & Algorithms* 38 (1-2), pp. 140–161. DOI: https://doi.org/10.1002/rsa.20354. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.20354.

Rumyantsev, Andrei and Alexander Shen (May 2014). "Probabilistic constructions of computable objects and a computable version of Lovász local lemma". In: *Fundamenta Informaticae* 132 (1), pp. 1–14. ISSN: 01692968. DOI: 10.3233/FI-2014-1029. URL: http://arxiv.org/abs/1305.1535.