

# Hypergraph Colouring Bounds using Probabilistic Methods

Chelsea Edmonds and Lawrence C. Paulson

February 6, 2026

## Abstract

This library includes several example applications of the probabilistic method for combinatorics to establish bounds for hypergraph colourings. This focuses on *Property B* — the existence of a two-colouring of the vertex set of a hypergraph. A stricter bound was formalised using the Lovász local lemma, which in turn required a surprisingly complex proof of the mutual independence principle for hypergraph edges that is often omitted on paper. The formalisation uncovered several interesting examples of circular intuition on proofs involving independence on paper. The formalisation is based on the textbook proofs from Alon and Spencer’s famous textbook, *The Probabilistic Method*[1], further supported by [3]. The mutual independence principle proof is inspired by the less precise proof provided in Molloy and Reed’s textbook on graph colourings [2], as it was omitted in all other sources. Additionally, this library demonstrates how locales can be used to establish a reusable probability space framework, thus minimizing the setup required for future formalisations requiring a probability space on numerous possible properties around an incidence system’s vertex set.

## Contents

<b>1</b>	<b>Hypergraph Colourings</b>	<b>2</b>
1.1	Function and Number extras . . . . .	2
1.2	Basic Definitions . . . . .	2
1.3	Monochromatic Edges . . . . .	4
1.4	Proper colourings . . . . .	4
1.5	n vertex colourings . . . . .	6
1.6	Alternate Partition Definition. . . . .	7
<b>2</b>	<b>Basic Probabilistic Method Application</b>	<b>9</b>
2.1	Probability Spaces for Incidence Set Systems . . . . .	9
2.2	More Hypergraph Colouring Results . . . . .	13
2.3	The Basic Application . . . . .	14

<b>3</b>	<b>Lovasz Local Framework Application</b>	<b>14</b>
3.1	More set extras . . . . .	14
3.2	Mutual Independence Principle for Hypergraphs . . . . .	15
3.3	Application Property B . . . . .	16
3.4	Application Corollary . . . . .	16

## 1 Hypergraph Colourings

**theory** *Hypergraph-Colourings* **imports** *Card-Partitions.Card-Partitions*  
*Hypergraph-Basics.Hypergraph-Variations HOL-Library.Extended-Real*  
*Girth-Chromatic.Girth-Chromatic-Misc*  
**begin**

### 1.1 Function and Number extras

**lemma** *surj-PiE*:  
**assumes**  $f \in A \rightarrow_E B$   
**assumes**  $f ' A = B$   
**assumes**  $b \in B$   
**obtains**  $a$  **where**  $a \in A$  **and**  $f a = b$   
*<proof>*

**lemma** *Stirling-gt-0*:  $n \geq k \implies k \neq 0 \implies \text{Stirling } n k > 0$   
*<proof>*

**lemma** *card-partition-on-ne*:  
**assumes**  $\text{card } A \geq n$   $n \neq 0$   
**shows**  $\{P. \text{partition-on } A P \wedge \text{card } P = n\} \neq \{\}$   
*<proof>*

**lemma** *enat-lt-INF*:  
**fixes**  $f :: 'a \Rightarrow \text{enat}$   
**assumes**  $(\text{INF } x \in S. f x) < t$   
**obtains**  $x$  **where**  $x \in S$  **and**  $f x < t$   
*<proof>*

### 1.2 Basic Definitions

**context** *hypergraph*  
**begin**

Edge colourings - using older partition approach

**definition** *edge-colouring* ::  $('a \text{ hyp-edge} \Rightarrow \text{colour}) \Rightarrow \text{colour set} \Rightarrow \text{bool}$  **where**  
 $\text{edge-colouring } f C \equiv \text{partition-on-mset } E \ \{\# \ \{\#h \in\# E . f h = c\# \} . c \in\#$   
 $(\text{mset-set } C)\#\}$

**definition** *proper-edge-colouring* ::  $('a \text{ hyp-edge} \Rightarrow \text{colour}) \Rightarrow \text{colour set} \Rightarrow \text{bool}$   
**where**

*proper-edge-colouring*  $f C \equiv \text{edge-colouring } f C \wedge$   
 $(\forall e1 e2 c. e1 \in \# E \wedge e2 \in \# E - \{\#e1\# \} \wedge c \in C \wedge f e1 = c \wedge f e2 = c$   
 $\longrightarrow e1 \cap e2 = \{\})$

A vertex colouring function with no edge monochromatic requirements

**abbreviation** *vertex-colouring*  $:: ('a \Rightarrow \text{colour}) \Rightarrow \text{nat} \Rightarrow \text{bool}$  **where**  
*vertex-colouring*  $f n \equiv f \in \mathcal{V} \rightarrow_E \{0..<n\}$

**lemma** *vertex-colouring-union*:

**assumes** *vertex-colouring*  $f n$

**shows**  $\bigcup \{\{v \in \mathcal{V}. f v = c\} \mid c. c \in \{0..<n\}\} = \mathcal{V}$

*<proof>*

**lemma** *vertex-colouring-disj*:

**assumes** *vertex-colouring*  $f n$

**assumes**  $p \in \{\{v \in \mathcal{V}. f v = c\} \mid c. c \in \{0..<n\}\}$

**assumes**  $p' \in \{\{v \in \mathcal{V}. f v = c\} \mid c. c \in \{0..<n\}\}$

**assumes**  $p \neq p'$

**shows**  $p \cap p' = \{\}$

*<proof>*

**lemma** *vertex-colouring-n0*:  $\mathcal{V} \neq \{\} \Longrightarrow \neg \text{vertex-colouring } f 0$

*<proof>*

**lemma** *vertex-colouring-image*: *vertex-colouring*  $f n \Longrightarrow v \in \mathcal{V} \Longrightarrow f v \in \{0..<n\}$

*<proof>*

**lemma** *vertex-colouring-image-edge-ss*: *vertex-colouring*  $f n \Longrightarrow e \in \# E \Longrightarrow f ' e \neq$   
 $e \subseteq \{0..<n\}$

*<proof>*

**lemma** *vertex-colour-edge-map-ne*: *vertex-colouring*  $f n \Longrightarrow e \in \# E \Longrightarrow f ' e \neq$   
 $\{\}$

*<proof>*

**lemma** *vertex-colouring-ne*: *vertex-colouring*  $f n \Longrightarrow f u \neq f v \Longrightarrow u \neq v$

*<proof>*

**lemma** *vertex-colour-one*:  $\mathcal{V} \neq \{\} \Longrightarrow \text{vertex-colouring } f 1 \Longrightarrow v \in \mathcal{V} \Longrightarrow f v =$   
 $(0::\text{nat})$

*<proof>*

**lemma** *vertex-colour-one-alt*:

**assumes**  $\mathcal{V} \neq \{\}$

**shows** *vertex-colouring*  $f (1::\text{nat}) \longleftrightarrow f = (\lambda v \in \mathcal{V}. 0::\text{nat})$

*<proof>*

**lemma** *vertex-colouring-partition*:

**assumes** *vertex-colouring*  $f n$

**assumes**  $f \text{ ' } \mathcal{V} = \{0..<n\}$   
**shows** *partition-on*  $\mathcal{V} \{ \{v \in \mathcal{V} . f v = c\} \mid c. c \in \{0..<n\} \}$   
*<proof>*

### 1.3 Monochromatic Edges

**definition** *mono-edge* ::  $('a \Rightarrow \text{colour}) \Rightarrow 'a \text{ hyp-edge} \Rightarrow \text{bool}$  **where**  
*mono-edge*  $f e \equiv \exists c. \forall v \in e. f v = c$

**lemma** *mono-edge-single*:  
**assumes**  $e \in \# E$   
**shows** *mono-edge*  $f e \iff \text{is-singleton} (f \text{ ' } e)$   
*<proof>*

**definition** *mono-edge-col* ::  $('a \Rightarrow \text{colour}) \Rightarrow 'a \text{ hyp-edge} \Rightarrow \text{colour} \Rightarrow \text{bool}$  **where**  
*mono-edge-col*  $f e c \equiv \forall v \in e. f v = c$

**lemma** *mono-edge-colI*:  $(\bigwedge v. v \in e \implies f v = c) \implies \text{mono-edge-col} f e c$   
*<proof>*

**lemma** *mono-edge-colD*:  $\text{mono-edge-col} f e c \implies (\bigwedge v. v \in e \implies f v = c)$   
*<proof>*

**lemma** *mono-edge-alt-col*:  $\text{mono-edge} f e \equiv \exists c. \text{mono-edge-col} f e c$   
*<proof>*

### 1.4 Proper colourings

A proper vertex colouring brings in the monochromatic edge decision. Note that this allows for a colouring of up to  $n$  colours, not precisely  $n$  colours

**definition** *is-proper-colouring* ::  $('a \Rightarrow \text{colour}) \Rightarrow \text{nat} \Rightarrow \text{bool}$  **where**  
*is-proper-colouring*  $f n \equiv \text{vertex-colouring} f n \wedge (\forall e \in \# E. \forall c \in \{0..<n\}. f \text{ ' } e \neq \{c\})$

**lemma** *is-proper-colouring-alt*:  $\text{is-proper-colouring} f n \iff \text{vertex-colouring} f n \wedge (\forall e \in \# E. \neg \text{is-singleton} (f \text{ ' } e))$   
*<proof>*

**lemma** *is-proper-colouring-alt2*:  $\text{is-proper-colouring} f n \iff \text{vertex-colouring} f n \wedge (\forall e \in \# E. \neg \text{mono-edge} f e)$   
*<proof>*

**lemma** *is-proper-colouringI[intro]*:  $\text{vertex-colouring} f n \implies (\bigwedge e . e \in \# E \implies \neg \text{is-singleton} (f \text{ ' } e)) \implies \text{is-proper-colouring} f n$   
*<proof>*

**lemma** *is-proper-colouringI2[intro]*:  $\text{vertex-colouring} f n \implies (\bigwedge e . e \in \# E \implies \neg \text{mono-edge} f e) \implies \text{is-proper-colouring} f n$

*<proof>*

**lemma** *is-proper-colouring-n0*:  $\mathcal{V} \neq \{\}$   $\implies \neg$  *is-proper-colouring f 0*  
*<proof>*

**lemma** *is-proper-colouring-empty*:  
**assumes**  $\mathcal{V} = \{\}$   
**shows** *is-proper-colouring f n*  $\longleftrightarrow f = (\lambda x . \text{undefined})$   
*<proof>*

**lemma** *is-proper-colouring-n1*:  
**assumes**  $\mathcal{V} \neq \{\}$   $E \neq \{\#\}$   
**shows**  $\neg$  *is-proper-colouring f 1*  
*<proof>*

**lemma** (in *fin-hypergraph*) *is-proper-colouring-image-card*:  
**assumes**  $\mathcal{V} \neq \{\}$   $E \neq \{\#\}$   
**assumes**  $n > 1$   
**assumes** *is-proper-colouring f n*  
**shows**  $\text{card } (f \text{ ' } \mathcal{V}) > 1$   
*<proof>*

More monochromatic edges

**lemma** *no-monochromatic-is-colouring*:  
**assumes**  $\forall e \in \# E . \neg$  *mono-edge f e*  
**assumes** *vertex-colouring f n*  
**shows** *is-proper-colouring f n*  
*<proof>*

**lemma** *ex-monochromatic-not-colouring*:  
**assumes**  $\exists e \in \# E .$  *mono-edge f e*  
**assumes** *vertex-colouring f n*  
**shows**  $\neg$  *is-proper-colouring f n*  
*<proof>*

**lemma** *mono-edge-colour-obtain*:  
**assumes** *mono-edge f e*  
**assumes** *vertex-colouring f n*  
**assumes**  $e \in \# E$   
**obtains** *c where*  $c \in \{0..<n\}$  **and** *mono-edge-col f e c*  
*<proof>*

Complete proper colourings - i.e. when n colours are required

**definition** *is-complete-proper-colouring*::  $(\text{'a} \Rightarrow \text{colour}) \Rightarrow \text{nat} \Rightarrow \text{bool}$  **where**  
*is-complete-proper-colouring f n*  $\equiv$  *is-proper-colouring f n*  $\wedge$   $f \text{ ' } \mathcal{V} = \{0..<n\}$

**lemma** *is-complete-proper-colouring-part*:  
**assumes** *is-complete-proper-colouring f n*  
**shows** *partition-on*  $\mathcal{V} \{ \{v \in \mathcal{V} . f v = c\} \mid c. c \in \{0..<n\}\}$

*<proof>*

**lemma** *is-complete-proper-colouring-n0*:  $\mathcal{V} \neq \{\}$   $\implies \neg$  *is-complete-proper-colouring*  $f$  0  
*<proof>*

**lemma** *is-complete-proper-colouring-n1*:  
**assumes**  $\mathcal{V} \neq \{\}$   $E \neq \{\#\}$   
**shows**  $\neg$  *is-complete-proper-colouring*  $f$  1  
*<proof>*

**lemma** (in *fin-hypergraph*) *is-proper-colouring-reduce*:  
**assumes** *is-proper-colouring*  $f$   $n$   
**obtains**  $f'$  **where** *is-complete-proper-colouring*  $f'$  ( $\text{card } (f \text{ ' } \mathcal{V})$ )  
*<proof>*

**lemma** (in *fin-hypergraph*) *two-colouring-is-complete*:  
**assumes**  $\mathcal{V} \neq \{\}$   
**assumes**  $E \neq \{\#\}$   
**assumes** *is-proper-colouring*  $f$  2  
**shows** *is-complete-proper-colouring*  $f$  2  
*<proof>*

## 1.5 $n$ vertex colourings

**definition** *is-n-colourable* ::  $\text{nat} \Rightarrow \text{bool}$  **where**  
*is-n-colourable*  $n \equiv \exists f . \text{is-proper-colouring } f$   $n$

**definition** *is-n-edge-colourable* ::  $\text{nat} \Rightarrow \text{bool}$  **where**  
*is-n-edge-colourable*  $n \equiv \exists f C . \text{card } C = n \longrightarrow \text{proper-edge-colouring } f$   $C$

**definition** *all-n-vertex-colourings* ::  $\text{nat} \Rightarrow ('a \Rightarrow \text{colour}) \text{ set}$  **where**  
*all-n-vertex-colourings*  $n \equiv \{f . \text{vertex-colouring } f$   $n\}$

**notation** *all-n-vertex-colourings* ( $\langle \mathcal{C}^n \rangle$ , [502] 500)

**lemma** *all-n-vertex-colourings-alt*:  $\mathcal{C}^n = \mathcal{V} \rightarrow_E \{0..<n\}$   
*<proof>*

**lemma** *vertex-colourings-empty*:  $\mathcal{V} \neq \{\}$   $\implies$  *all-n-vertex-colourings* 0 =  $\{\}$   
*<proof>*

**lemma** (in *fin-hypergraph*) *vertex-colourings-fin* : *finite* ( $\mathcal{C}^n$ )  
*<proof>*

**lemma** (in *fin-hypergraph*) *count-vertex-colourings*:  $\text{card } (\mathcal{C}^n) = n \wedge \text{horder}$   
*<proof>*

**lemma** *vertex-colourings-nempty*:

**assumes**  $\text{card } \mathcal{V} \geq n$   
**assumes**  $n \neq 0$   
**shows**  $\mathcal{C}^n \neq \{\}$   
 $\langle \text{proof} \rangle$

**lemma** *vertex-colourings-one*:  
**assumes**  $\mathcal{V} \neq \{\}$   
**shows**  $\mathcal{C}^1 = \{\lambda \ v \in \mathcal{V} . 0\}$   
 $\langle \text{proof} \rangle$

**lemma** *mono-edge-set-union*:  
**assumes**  $e \in \# E$   
**shows**  $\{f \in \mathcal{C}^n . \text{mono-edge } f \ e\} = (\bigcup c \in \{0..<n\} . \{f \in \mathcal{C}^n . \text{mono-edge-col } f \ e \ c\})$   
 $\langle \text{proof} \rangle$

**end**

Property B set up

**abbreviation** (in *hypergraph*) *has-property-B* :: *bool* **where**  
*has-property-B*  $\equiv$  *is-n-colourable* 2

**abbreviation** *hyp-graph-order*:: 'a *hyp-graph*  $\Rightarrow$  *nat* **where**  
*hyp-graph-order*  $h \equiv$  *card* (*hyp-verts*  $h$ )

**definition** *not-col-n-uni-hyps*:: *nat*  $\Rightarrow$  'a *hyp-graph set*  
**where** *not-col-n-uni-hyps*  $n \equiv$   $\{ h . \text{fin-kuniform-hypergraph-nt } (\text{hyp-verts } h) (\text{hyp-edges } h) \ n \ \wedge$   
 $\neg (\text{hypergraph.has-property-B } (\text{hyp-verts } h) (\text{hyp-edges } h)) \}$

**definition** *min-edges-colouring* :: *nat*  $\Rightarrow$  'a *itself*  $\Rightarrow$  *enat* **where**  
*min-edges-colouring*  $n \equiv$  *INF*  $h \in ((\text{not-col-n-uni-hyps } n) :: \text{'a hyp-graph set}) .$   
*enat* (*size* (*hyp-edges*  $h$ ))

**lemma** *obtains-min-edge-colouring*:  
**fixes**  $z :: \text{'a itself}$   
**assumes** *min-edges-colouring*  $n \ z < x$   
**obtains**  $h :: \text{'a hyp-graph}$  **where**  $h \in \text{not-col-n-uni-hyps } n$  **and** *enat* (*size* (*hyp-edges*  $h$ ))  $< x$   
 $\langle \text{proof} \rangle$

## 1.6 Alternate Partition Definition.

Note that the indexed definition should be used most of the time instead

**context** *hypergraph*  
**begin**

**definition** *is-proper-colouring-part* :: 'a *set set*  $\Rightarrow$  *bool* **where**  
*is-proper-colouring-part*  $C \equiv$  *partition-on*  $\mathcal{V} \ C \ \wedge \ (\forall c \in C . \forall e \in \# E . \neg e \subseteq c)$

**definition** *is-n-colourable-part* :: nat  $\Rightarrow$  bool **where**  
*is-n-colourable-part* n  $\equiv \exists C . \text{card } C = n \longrightarrow \text{is-proper-colouring-part } C$

**abbreviation** *has-property-B-part* :: bool **where**  
*has-property-B-part*  $\equiv \text{is-n-colourable-part } 2$

**definition** *mono-edge-ss* :: 'a set set  $\Rightarrow$  'a hyp-edge  $\Rightarrow$  bool **where**  
*mono-edge-ss* C e  $\equiv \exists c \in C . e \subseteq c$

**lemma** *is-proper-colouring-partI*: *partition-on*  $\mathcal{V} C \Longrightarrow (\forall c \in C . \forall e \in \# E . \neg e \subseteq c) \Longrightarrow$   
*is-proper-colouring-part* C  
 <proof>

**lemma** *no-monochromatic-is-colouring-part*:  
**assumes**  $\forall e \in \# E . \neg \text{mono-edge-ss } C e$   
**assumes** *partition-on*  $\mathcal{V} C$   
**shows** *is-proper-colouring-part* C  
 <proof>

**lemma** *ex-monochromatic-not-colouring-part*:  
**assumes**  $\exists e \in \# E . \text{mono-edge-ss } C e$   
**assumes** *partition-on*  $\mathcal{V} C$   
**shows**  $\neg \text{is-proper-colouring-part } C$   
 <proof>

**definition** *all-n-vertex-colourings-part* :: nat  $\Rightarrow$  'a set set set **where**  
*all-n-vertex-colourings-part* n  $\equiv \{C . \text{partition-on } \mathcal{V} C \wedge \text{card } C = n\}$

**lemma** (in *fin-hypergraph*) *all-vertex-colourings-part-fin*: *finite* (*all-n-vertex-colourings-part* n)  
 <proof>

**lemma** *all-vertex-colourings-part-nempty*:  $\text{card } \mathcal{V} \geq n \Longrightarrow n \neq 0 \Longrightarrow \text{all-n-vertex-colourings-part } n \neq \{\}$   
 <proof>

**lemma** *disjoint-family-on-colourings*:  
**assumes**  $e \in \# E$   
**shows** *disjoint-family-on*  $(\lambda c . \{f \in \mathcal{C}^n . \text{mono-edge-col } f e c\}) \{0..<n\}$   
 <proof>

**end**

**end**

## 2 Basic Probabilistic Method Application

This section establishes step (1) of the basic framework for incidence set systems, as well as some basic bounds on hypergraph colourings

**theory** *Basic-Bounds-Application* **imports** *Lovasz-Local.Basic-Method-Hypergraph-Colourings*  
**begin**

### 2.1 Probability Spaces for Incidence Set Systems

This is effectively step (1) of the formal framework for probabilistic method. Unlike stages (3) and (4), which were formalised in the *Lovasz\_Local\_Lemma* AFP entry, this stage required a formalisation of incidence set systems as well as the background probability space locales

A basic probability space for a point measure on a non-trivial structure

**locale** *vertex-fn-space* = *fin-hypersystem-vne* +  
**fixes**  $F :: 'a \text{ set} \Rightarrow 'b \text{ set}$   
**fixes**  $p :: 'b \Rightarrow \text{real}$   
**assumes**  $ne: F \mathcal{V} \neq \{\}$   
**assumes**  $fin: \text{finite } (F \mathcal{V})$   
**assumes**  $pgte0: \bigwedge fv . fv \in F \mathcal{V} \Longrightarrow p \text{ } fv \geq 0$   
**assumes**  $sump: (\sum x \in (F \mathcal{V}) . p \text{ } x) = 1$   
**begin**

**definition**  $\Omega \equiv F \mathcal{V}$

**lemma** *fin- $\Omega$* :  $\text{finite } \Omega$   
*<proof>*

**lemma** *ne- $\Omega$* :  $\Omega \neq \{\}$   
*<proof>*

**definition**  $M = \text{point-measure } \Omega \text{ } p$

**lemma** *space-eq*:  $\text{space } M = \Omega$   
*<proof>*

**lemma** *sets-eq*:  $\text{sets } M = \text{Pow } (\Omega)$   
*<proof>*

**lemma** *finite-event*:  $A \subseteq \Omega \Longrightarrow \text{finite } A$   
*<proof>*

**lemma** *emeasure-eq*:  $\text{emeasure } M \text{ } A = (\text{if } (A \subseteq \Omega) \text{ then } (\sum a \in A. p \text{ } a) \text{ else } 0)$   
*<proof>*

**lemma** *integrable-M*[*intro, simp*]:  $\text{integrable } M \text{ } (f::- \Rightarrow \text{real})$   
*<proof>*

**lemma** *borel-measurable-M*[*measurable*]:  $f \in \text{borel-measurable } M$   
 ⟨*proof*⟩

**lemma** *prob-space-M*: *prob-space*  $M$   
 ⟨*proof*⟩

**end**

**sublocale** *vertex-fn-space*  $\subseteq$  *prob-space*  $M$   
 ⟨*proof*⟩

A uniform variation of the space

**locale** *vertex-fn-space-uniform* = *fin-hypersystem-vne* +

**fixes**  $F :: 'a \text{ set} \Rightarrow 'b \text{ set}$

**assumes**  $ne: F \mathcal{V} \neq \{\}$

**assumes**  $fin: \text{finite } (F \mathcal{V})$

**begin**

**definition**  $\Omega U \equiv F \mathcal{V}$

**definition**  $MU \equiv \text{uniform-count-measure } \Omega U$

**end**

**sublocale** *vertex-fn-space-uniform*  $\subseteq$  *vertex-fn-space*  $\mathcal{V} E F (\lambda x. 1 / \text{card } \Omega U)$

**rewrites**  $\Omega = \Omega U$  **and**  $M = MU$

⟨*proof*⟩

**context** *vertex-fn-space-uniform*

**begin**

**lemma** *emeasure-eq*:  $\text{emeasure } MU A = (\text{if } (A \subseteq \Omega U) \text{ then } ((\text{card } A) / \text{card } (\Omega U))$   
*else*  $0$ )

⟨*proof*⟩

**lemma** *measure-eq-valid*:  $A \in \text{events} \implies \text{measure } MU A = (\text{card } A) / \text{card } (\Omega U)$

⟨*proof*⟩

**lemma** *expectation-eq*:

**shows**  $\text{expectation } f = (\sum x \in \Omega U. f x) / \text{card } \Omega U$

⟨*proof*⟩

**end**

A probability space over the full vertex set

**locale** *vertex-space* = *fin-hypersystem-vne* +

**fixes**  $p :: 'a \Rightarrow \text{real}$

**assumes**  $pgte0: \bigwedge fv. fv \in \mathcal{V} \implies p fv \geq 0$

**assumes** *sump*:  $(\sum x \in (\mathcal{V}) . p x) = 1$

**sublocale** *vertex-space*  $\subseteq$  *vertex-fn-space*  $\mathcal{V} E \lambda i . i p$   
**rewrites**  $\Omega = \mathcal{V}$   
*<proof>*

A uniform variation of the probability space over the vertex set

**locale** *vertex-space-uniform* = *fin-hypersystem-vne*

**sublocale** *vertex-space-uniform*  $\subseteq$  *vertex-fn-space-uniform*  $\mathcal{V} E \lambda i . i$   
**rewrites**  $\Omega U = \mathcal{V}$   
*<proof>*

A uniform probability space over a vertex subset

**locale** *vertex-ss-space-uniform* = *fin-hypersystem-vne* +  
**fixes** *VS*  
**assumes** *vs-ss*:  $VS \subseteq \mathcal{V}$   
**assumes** *ne-vs*:  $VS \neq \{\}$   
**begin**

**lemma** *finite-vs: finite VS*  
*<proof>*

**end**

**sublocale** *vertex-ss-space-uniform*  $\subseteq$  *vertex-fn-space-uniform*  $\mathcal{V} E \lambda i . VS$   
**rewrites**  $\Omega = VS$   
*<proof>*

A non-uniform prob space over a vertex subset

**locale** *vertex-ss-space* = *fin-hypersystem-vne* +  
**fixes** *VS*  
**assumes** *vs-ss*:  $VS \subseteq \mathcal{V}$   
**assumes** *ne-vs*:  $VS \neq \{\}$   
**fixes** *p* :: '*a*  $\Rightarrow$  *real*  
**assumes** *pgte0*:  $\bigwedge fv . fv \in VS \implies p fv \geq 0$   
**assumes** *sump*:  $(\sum x \in (VS) . p x) = 1$   
**begin**

**lemma** *finite-vs: finite VS*  
*<proof>*

**end**

**sublocale** *vertex-ss-space*  $\subseteq$  *vertex-fn-space*  $\mathcal{V} E \lambda i . VS p$   
**rewrites**  $\Omega = VS$   
*<proof>*

A uniform probability space over a property on the vertex set

**locale** *vertex-prop-space* = *fin-hypersystem-vne* +  
**fixes**  $P :: 'b \text{ set}$   
**assumes** *finP*: *finite P*  
**assumes** *nempty-P*:  $P \neq \{\}$

**sublocale** *vertex-prop-space*  $\subseteq$  *vertex-fn-space-uniform*  $\mathcal{V} E \lambda V. V \rightarrow_E P$   
**rewrites**  $\Omega U = \mathcal{V} \rightarrow_E P$   
 $\langle$ *proof* $\rangle$

**context** *vertex-prop-space*  
**begin**

**lemma** *prob-uniform-vertex-subset*:  
**assumes**  $b \in P$   
**assumes**  $d \subseteq \mathcal{V}$   
**shows** *prob*  $\{f \in \Omega . (\forall v \in d . f v = b)\} = 1 / ((\text{card } P) \text{ powi } (\text{card } d))$   
 $\langle$ *proof* $\rangle$

**lemma** *prob-uniform-vertex*:  
**assumes**  $b \in P$   
**assumes**  $v \in \mathcal{V}$   
**shows** *prob*  $\{f \in \Omega U . f v = b\} = 1 / (\text{card } P)$   
 $\langle$ *proof* $\rangle$

**end**

A uniform vertex colouring space

**locale** *vertex-colour-space* = *fin-hypergraph-nt* +  
**fixes**  $n :: \text{nat}$   
**assumes** *n-lt-order*:  $n \leq \text{horder}$   
**assumes** *n-not-zero*:  $n \neq 0$

**sublocale** *vertex-colour-space*  $\subseteq$  *vertex-prop-space*  $\mathcal{V} E \{0..<n\}$   
**rewrites**  $\Omega U = \mathcal{C}^n$   
 $\langle$ *proof* $\rangle$

This probability space contains several useful lemmas on basic vertex colouring probabilities (and monochromatic edges), which are facts that are typically either not proven, or have very short proofs on paper

**context** *vertex-colour-space*  
**begin**

**lemma** *colour-set-event*:  $\{f \in \mathcal{C}^n . \text{mono-edge-col } f e c\} \in \text{events}$   
 $\langle$ *proof* $\rangle$

**lemma** *colour-functions-event*:  $(\lambda c. \{f \in \mathcal{C}^n . \text{mono-edge-col } f e c\}) ' \{0..<n\} \subseteq \text{events}$   
 $\langle$ *proof* $\rangle$

**lemma** *prob-vertex-colour*:  $v \in \mathcal{V} \implies c \in \{0..<n\} \implies \text{prob} \{f \in \mathcal{C}^n . f v = c\} = 1/n$   
 ⟨proof⟩

**lemma** *prob-edge-colour*:  
**assumes**  $e \in \# E$   $c \in \{0..<n\}$   
**shows**  $\text{prob} \{f \in \mathcal{C}^n . \text{mono-edge-col } f e c\} = 1/(n \text{ powi } (\text{card } e))$   
 ⟨proof⟩

**lemma** *prob-monochromatic-edge-inv*:  
**assumes**  $e \in \# E$   
**shows**  $\text{prob} \{f \in \mathcal{C}^n . \text{mono-edge } f e\} = 1/(n \text{ powi } (\text{int } (\text{card } e) - 1))$   
 ⟨proof⟩

**lemma** *prob-monochromatic-edge*:  
**assumes**  $e \in \# E$   
**shows**  $\text{prob} \{f \in \mathcal{C}^n . \text{mono-edge } f e\} = n \text{ powi } (1 - \text{int } (\text{card } e))$   
 ⟨proof⟩

**lemma** *prob-monochromatic-edge-bound*:  
**assumes**  $e \in \# E$   
**assumes**  $\bigwedge e. e \in \# E \implies \text{card } e \geq k$   
**assumes**  $k > 0$   
**shows**  $\text{prob} \{f \in \mathcal{C}^n . \text{mono-edge } f e\} \leq 1/((\text{real } n) \text{ powi } (k-1))$   
 ⟨proof⟩

end

## 2.2 More Hypergraph Colouring Results

**context** *fin-hypergraph-nt*  
**begin**

**lemma** *not-proper-colouring-edge-mono*:  $\{f \in \mathcal{C}^n . \neg \text{is-proper-colouring } f n\} = (\bigcup e \in (\text{set-mset } E). \{f \in \mathcal{C}^n . \text{mono-edge } f e\})$   
 ⟨proof⟩

**lemma** *proper-colouring-edge-mono*:  $\{f \in \mathcal{C}^n . \text{is-proper-colouring } f n\} = (\bigcap e \in (\text{set-mset } E). \{f \in \mathcal{C}^n . \neg \text{mono-edge } f e\})$   
 ⟨proof⟩

**lemma** *proper-colouring-edge-mono-compl*:  $\{f \in \mathcal{C}^n . \text{is-proper-colouring } f n\} = (\bigcap e \in (\text{set-mset } E). \mathcal{C}^n - \{f \in \mathcal{C}^n . \text{mono-edge } f e\})$   
 ⟨proof⟩

**lemma** *event-is-proper-colouring*:  
**assumes**  $g \in \mathcal{C}^n$   
**assumes**  $g \notin (\bigcup e \in (\text{set-mset } E). \{f \in \mathcal{C}^n . \text{mono-edge } f e\})$

**shows** *is-proper-colouring*  $g\ n$   
 $\langle$ *proof* $\rangle$

**end**

## 2.3 The Basic Application

The comments below show the basic framework steps

**context** *fin-kuniform-hypergraph-nt*

**begin**

**proposition** *erdos-propertyB*:

**assumes** *size*  $E < (2^{k-1})$

**assumes**  $k > 0$

**shows** *has-property-B*

$\langle$ *proof* $\rangle$

**end**

**corollary** *erdos-propertyB-min*:

**fixes**  $z :: 'a$  *itself*

**assumes**  $n > 0$

**shows** (*min-edges-colouring*  $n\ z$ )  $\geq 2^{n-1}$

$\langle$ *proof* $\rangle$

**end**

## 3 Lovasz Local Framework Application

**theory** *LLL-Applications* **imports** *Lovasz-Local.Lovasz-Local-Lemma*

*Lovasz-Local.Indep-Events Twelvefold-Way.Twelvefold-Way-Core*

*Design-Theory.Multisets-Extras Basic-Bounds-Application*

**begin**

### 3.1 More set extras

**lemma** *multiset-remove1-filter*:  $a \in\# A \implies P\ a \implies$

$\{\#b \in\# A . P\ b\#\} = \{\#b \in\# \text{remove1-mset } a\ A . P\ b\#\} + \{\#a\#\}$

$\langle$ *proof* $\rangle$

**lemma** *card-partition-image*:

**assumes** *finite*  $C$

**assumes** *finite*  $(\bigcup c \in C . f\ c)$

**assumes**  $(\bigwedge c . c \in C \implies \text{card } (f\ c) = k)$

**assumes**  $(\bigwedge c1\ c2 . c1 \in C \implies c2 \in C \implies c1 \neq c2 \implies f\ c1 \cap f\ c2 = \{\})$

**shows**  $k * \text{card } (f\ 'C) = \text{card } (\bigcup c \in C . f\ c)$

$\langle$ *proof* $\rangle$

**lemma** *mset-set-implies*:

**assumes** *image-mset*  $f$  (*mset-set*  $A$ ) =  $B$

**assumes**  $\bigwedge a . a \in A \implies P (f a)$

**shows**  $\bigwedge b . b \in \# B \implies P b$

*<proof>*

**lemma** *card-partition-image-inj*:

**assumes** *finite*  $C$

**assumes** *inj-on*  $f C$

**assumes** *finite*  $(\bigcup c \in C . f c)$

**assumes**  $(\bigwedge c . c \in C \implies \text{card} (f c) = k)$

**assumes**  $(\bigwedge c1 c2 . c1 \in C \implies c2 \in C \implies c1 \neq c2 \implies f c1 \cap f c2 = \{\})$

**shows**  $k * \text{card} (C) = \text{card} (\bigcup c \in C . f c)$

*<proof>*

**lemma** *size-big-union-sum2*:

**fixes**  $M :: 'a \Rightarrow 'b$  *multiset*

**shows**  $\text{size} (\sum x \in \# X . M x) = (\sum x \in \# X . \text{size} (M x))$

*<proof>*

**lemma** *size-big-union-sum2-const*:

**fixes**  $M :: 'a \Rightarrow 'b$  *multiset*

**assumes**  $\bigwedge x . x \in \# X \implies \text{size} (M x) = k$

**shows**  $\text{size} (\sum x \in \# X . M x) = \text{size} X * k$

*<proof>*

**lemma** *count-sum-mset2*:  $\text{count} (\sum x \in \# X . M x) a = (\sum x \in \# X . \text{count} (M x) a)$

*<proof>*

**lemma** *mset-subset-eq-lem1*:

$(\bigwedge a . a \in \# A \implies \text{count} A a \leq \text{count} B a) \implies A \subseteq \# B$

*<proof>*

**lemma** *mset-obtain-from-filter*:

**assumes**  $a \in \# \{\# b \in \# B . P b \#\}$

**shows**  $a \in \# B$  **and**  $P a$

*<proof>*

## 3.2 Mutual Independence Principle for Hypergraphs

**context** *fin-hypergraph-nt*

**begin**

**definition** (**in** *incidence-system*) *block-intersect-count* ::  $'a$  *set*  $\Rightarrow$  *nat* **where**  
*block-intersect-count*  $b \equiv \text{size} \{\# b2 \in \# (B - \{\# b \#\}) . b2 \cap b \neq \{\} \#\}$

**lemma** (**in** *hypergraph*) *edge-intersect-count-inc*:

**assumes**  $e \in \# E$

**shows**  $\text{size } \{ \# f \in \# E . f \cap e \neq \{ \} \# \} = \text{block-intersect-count } e + 1$   
 <proof>

**lemma** *disjoint-set-is-mutually-independent*:

**assumes** *iin*:  $i \in \{ 0..<(\text{size } E) \}$

**assumes** *idfn*:  $\text{idf} \in \{ 0..<\text{size } E \} \rightarrow_E \text{set-mset } E$

**assumes** *Aefn*:  $\bigwedge i. i \in \{ 0..<\text{size } E \} \implies \text{Ae } i = \{ f \in \mathcal{C}^2 . \text{mono-edge } f (\text{idf } i) \}$

**shows** *prob-space.mutual-indep-events* (*uniform-count-measure* ( $\mathcal{C}^2$ )) (*Ae* *i*) *Ae*  
 ( $\{ j \in \{ 0..<(\text{size } E) \} . (\text{idf } j \cap \text{idf } i) = \{ \} \}$ )

<proof>

**lemma** *intersect-empty-set-size*:

**assumes**  $\bigwedge e . e \in \# E \implies \text{size } \{ \# f \in \# (E - \{ \# e \# \}) . f \cap e \neq \{ \} \# \} \leq d$

**assumes**  $e2 \in \# E$

**shows**  $\text{size } \{ \# e \in \# E . e \cap e2 = \{ \} \# \} \geq \text{size } E - d - 1$  (**is**  $\text{size } ?S' \geq \text{size } E - d - 1$ )

<proof>

### 3.3 Application Property B

Probabilistic framework clearly notated

**proposition** *erdos-propertyB-LLL*:

**assumes**  $\bigwedge e . e \in \# E \implies \text{card } e \geq k$

**assumes**  $\bigwedge e . e \in \# E \implies \text{size } \{ \# f \in \# (E - \{ \# e \# \}) . f \cap e \neq \{ \} \# \} \leq d$

**assumes**  $\exp(1) * (d+1) \leq (2 \text{ powi } (k - 1))$

**assumes**  $k > 0$

**shows** *has-property-B*

<proof>

end

### 3.4 Application Corollary

A corollary on hypergraphs where  $k \geq 9$

**lemma** *exp-ineq-k9*:

**fixes**  $k :: \text{nat}$

**assumes**  $k \geq 9$

**shows**  $\exp(1) * (k * (k - 1) + 1) < 2^{\sim}(k-1)$

<proof>

**context** *fin-kuniform-regular-hypgraph-nt*

**begin**

Good example of a combinatorial counting proof in a formal environment

**lemma** (**in** *fin-dregular-hypergraph*) *hdeg-remove-one*:

**assumes**  $e \in \# E$

**assumes**  $v \in \# \text{mset-set } e$

**shows**  $\text{size } \{ \# f \in \# (E - \{ \# e \# \}) . v \in f \# \} = d - 1$

*<proof>*

**lemma** *max-intersecting-edges*:

**assumes**  $e \in \# E$

**shows**  $\text{size } \{ \# f \in \# (E - \{ \# e \# \}) . f \cap e \neq \{ \# \} \} \leq k * (k - 1)$

*<proof>*

**corollary** *erdos-propertyB-LLL9*:

**assumes**  $k \geq 9$

**shows** *has-property-B*

*<proof>*

**end**

**end**

**theory** *Hypergraph-Colourings-Root*

**imports**

*Hypergraph-Colourings*

*Basic-Bounds-Application*

*LLL-Applications*

**begin**

**end**

## References

- [1] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, Hoboken, N.J, 4th edition, 2016.
- [2] M. Molloy and B. Reed. *Graph Colouring and the Probabilistic Method*. Algorithms and Combinatorics. Springer, 2002.
- [3] Y. Zhao. Probabilistic methods in combinatorics, 2020. Lecture notes MIT 18.226, Fall 2020, [https://ocw.mit.edu/courses/18-226-probabilistic-method-in-combinatorics-fall-2020/resources/mit18\\_226f20\\_full\\_notes/](https://ocw.mit.edu/courses/18-226-probabilistic-method-in-combinatorics-fall-2020/resources/mit18_226f20_full_notes/).