

# Lovasz Local Lemma

Chelsea Edmonds and Lawrence C. Paulson

September 22, 2023

## Abstract

This entry aims to formalise several useful general techniques for using the *probabilistic method* for combinatorial structures (or discrete spaces more generally). In particular, it focuses on bounding tools, such as the union and complete independence bounds, and the first formalisation of the pivotal Lovász local lemma. The formalisation focuses on the general lemma, however also proves several useful variations, including the more well known symmetric version. Both the original formalisation and several of the variations used dependency graphs, which were formalised using Noschinski’s general directed graph library [2]. Additionally, the entry provides several useful existence lemmas, required at the end of most probabilistic proofs on combinatorial structures. Finally, the entry includes several significant extensions to the existing probability libraries, particularly for conditional probability (such as Bayes theorem) and independent events. The formalisation is primarily based on Alon and Spencer’s textbook [1], as well as Zhao’s course notes [3].

## Contents

<b>1</b>	<b>Extensional function extras</b>	<b>2</b>
1.1	Relations and Extensional Function sets . . . . .	2
1.2	Cardinality Lemmas . . . . .	4
<b>2</b>	<b>Digraph extensions</b>	<b>4</b>
<b>3</b>	<b>General Event Lemmas</b>	<b>5</b>
<b>4</b>	<b>Conditional Probability Library Extensions</b>	<b>9</b>
4.1	Miscellaneous Set and List Lemmas . . . . .	9
4.2	Conditional Probability Basics . . . . .	9
4.3	Bayes Theorem . . . . .	11
4.4	Conditional Probability Multiplication Rule . . . . .	12

<b>5</b>	<b>Independent Events</b>	<b>17</b>
5.1	More bijection helpers . . . . .	17
5.2	Independent Event Extensions . . . . .	18
5.3	Mutual Independent Events . . . . .	24
<b>6</b>	<b>The Basic Probabilistic Method Framework</b>	<b>29</b>
6.1	More Set and Multiset lemmas . . . . .	29
6.2	Existence Lemmas . . . . .	30
6.3	Basic Bounds . . . . .	31
<b>7</b>	<b>Lovasz Local Lemma</b>	<b>33</b>
7.1	Random Lemmas on Product Operator . . . . .	33
7.2	Dependency Graph Concept . . . . .	34
7.3	Lovasz Local General Lemma . . . . .	35
7.4	Lovasz Corollaries and Variations . . . . .	36

## 1 Extensional function extras

Counting lemmas (i.e. reasoning on cardinality) of sets on the extensional function relation

```
theory PiE-Rel-Extras imports Card-Partitions.Card-Partitions
begin
```

### 1.1 Relations and Extensional Function sets

A number of lemmas to convert between relations and functions for counting purposes. Note, ultimately not needed in this formalisation, but may be of use in the future

```
lemma Range-unfold: Range r = {y.  $\exists x. (x, y) \in r$ }
  <proof>
```

```
definition fun-to-rel:: 'a set  $\Rightarrow$  'b set  $\Rightarrow$  ('a  $\Rightarrow$  'b)  $\Rightarrow$  ('a  $\times$  'b) set where
fun-to-rel A B f  $\equiv$  {(a, b) | a b . a  $\in$  A  $\wedge$  b  $\in$  B  $\wedge$  f a = b}
```

```
definition rel-to-fun:: ('a  $\times$  'b) set  $\Rightarrow$  ('a  $\Rightarrow$  'b) where
rel-to-fun R  $\equiv$   $\lambda a .$  (if a  $\in$  Domain R then (THE b . (a, b)  $\in$  R) else undefined)
```

```
lemma fun-to-relI: a  $\in$  A  $\Longrightarrow$  b  $\in$  B  $\Longrightarrow$  f a = b  $\Longrightarrow$  (a, b)  $\in$  fun-to-rel A B f
  <proof>
```

```
lemma fun-to-rel-alt: fun-to-rel A B f  $\equiv$  {(a, f a) | a b . a  $\in$  A  $\wedge$  f a  $\in$  B}
  <proof>
```

```
lemma fun-to-relI2: a  $\in$  A  $\Longrightarrow$  f a  $\in$  B  $\Longrightarrow$  (a, f a)  $\in$  fun-to-rel A B f
  <proof>
```

**lemma** *rel-to-fun-in[simp]*:  $a \in \text{Domain } R \implies (\text{rel-to-fun } R) a = (\text{THE } b . (a, b) \in R)$

*<proof>*

**lemma** *rel-to-fun-undefined[simp]*:  $a \notin \text{Domain } R \implies (\text{rel-to-fun } R) a = \text{undefined}$

*<proof>*

**lemma** *single-valued-unique-Dom-iff*:  $\text{single-valued } R \iff (\forall x \in \text{Domain } R. \exists! y . (x, y) \in R)$

*<proof>*

**lemma** *rel-to-fun-range*:

**assumes** *single-valued*  $R$

**assumes**  $a \in \text{Domain } R$

**shows**  $(\text{THE } b . (a, b) \in R) \in \text{Range } R$

*<proof>*

**lemma** *rel-to-fun-extensional*:  $\text{single-valued } R \implies \text{rel-to-fun } R \in (\text{Domain } R \rightarrow_E \text{Range } R)$

*<proof>*

**lemma** *single-value-fun-to-rel*:  $\text{single-valued } (\text{fun-to-rel } A B f)$

*<proof>*

**lemma** *fun-to-rel-domain*:

**assumes**  $f \in A \rightarrow_E B$

**shows**  $\text{Domain } (\text{fun-to-rel } A B f) = A$

*<proof>*

**lemma** *fun-to-rel-range*:

**assumes**  $f \in A \rightarrow_E B$

**shows**  $\text{Range } (\text{fun-to-rel } A B f) \subseteq B$

*<proof>*

**lemma** *rel-to-fun-to-rel*:

**assumes**  $f \in A \rightarrow_E B$

**shows**  $\text{rel-to-fun } (\text{fun-to-rel } A B f) = f$

*<proof>*

**lemma** *fun-to-rel-to-fun*:

**assumes** *single-valued*  $R$

**shows**  $\text{fun-to-rel } (\text{Domain } R) (\text{Range } R) (\text{rel-to-fun } R) = R$

*<proof>*

**lemma** *bij-betw-fun-to-rel*:

**assumes**  $f \in A \rightarrow_E B$

**shows** *bij-betw*  $(\lambda a . (a, f a)) A (\text{fun-to-rel } A B f)$

*<proof>*

**lemma** *fun-to-rel-indiv-card*:  
**assumes**  $f \in A \rightarrow_E B$   
**shows**  $\text{card } (\text{fun-to-rel } A B f) = \text{card } A$   
 $\langle \text{proof} \rangle$

**lemma** *fun-to-rel-inj*:  
**assumes**  $C \subseteq A \rightarrow_E B$   
**shows** *inj-on*  $(\text{fun-to-rel } A B) C$   
 $\langle \text{proof} \rangle$

**lemma** *fun-to-rel-ss*:  $\text{fun-to-rel } A B f \subseteq A \times B$   
 $\langle \text{proof} \rangle$

**lemma** *card-fun-to-rel*:  $C \subseteq A \rightarrow_E B \implies \text{card } C = \text{card } ((\lambda f . \text{fun-to-rel } A B f) ` C)$   
 $\langle \text{proof} \rangle$

## 1.2 Cardinality Lemmas

Lemmas to count variations of filtered sets over the extensional function set relation

**lemma** *card-PiE-filter-range-set*:  
**assumes**  $\bigwedge a. a \in A' \implies X a \in C$   
**assumes**  $A' \subseteq A$   
**assumes** *finite*  $A$   
**shows**  $\text{card } \{f \in A \rightarrow_E C . \forall a \in A' . f a = X a\} = (\text{card } C) \frown (\text{card } A - \text{card } A')$   
 $\langle \text{proof} \rangle$

**lemma** *card-PiE-filter-range-indiv*:  $X a' \in C \implies a' \in A \implies \text{finite } A \implies$   
 $\text{card } \{f \in A \rightarrow_E C . f a' = X a'\} = (\text{card } C) \frown (\text{card } A - 1)$   
 $\langle \text{proof} \rangle$

**lemma** *card-PiE-filter-range-set-const*:  $c \in C \implies A' \subseteq A \implies \text{finite } A \implies$   
 $\text{card } \{f \in A \rightarrow_E C . \forall a \in A' . f a = c\} = (\text{card } C) \frown (\text{card } A - \text{card } A')$   
 $\langle \text{proof} \rangle$

**lemma** *card-PiE-filter-range-set-nat*:  $c \in \{0..<n\} \implies A' \subseteq A \implies \text{finite } A \implies$   
 $\text{card } \{f \in A \rightarrow_E \{0..<n\} . \forall a \in A' . f a = c\} = n \frown (\text{card } A - \text{card } A')$   
 $\langle \text{proof} \rangle$

**end**

## 2 Digraph extensions

Extensions to the existing library for directed graphs, basically neighborhood

**theory** *Digraph-Extensions*  
**imports**

```

    Graph-Theory.Digraph
    Graph-Theory.Pair-Digraph
begin

definition (in pre-digraph) neighborhood :: 'a ⇒ 'a set where
neighborhood u ≡ {v ∈ verts G . dominates G u v}

lemma (in wf-digraph) neighborhood-wf: neighborhood v ⊆ verts G
⟨proof⟩

lemma (in pair-pre-digraph) neighborhood-alt:
neighborhood u = {v ∈ pverts G . (u, v) ∈ parcs G}
⟨proof⟩

lemma (in fin-digraph) neighborhood-finite: finite (neighborhood v)
⟨proof⟩

lemma (in wf-digraph) neighborhood-edge-iff: y ∈ neighborhood x ⟷ (x, y) ∈
arcs-ends G
⟨proof⟩

lemma (in loopfree-digraph) neighborhood-self-not: v ∉ (neighborhood v)
⟨proof⟩

lemma (in nomulti-digraph) inj-on-head-out-arcs: inj-on (head G) (out-arcs G u)
⟨proof⟩

lemma (in nomulti-digraph) out-degree-neighborhood: out-degree G u = card (neighborhood
u)
⟨proof⟩

lemma (in digraph) neighborhood-empty-iff: out-degree G u = 0 ⟷ neighborhood
u = {}
⟨proof⟩

end

```

### 3 General Event Lemmas

General lemmas for reasoning on events in probability spaces after different operations

```

theory Prob-Events-Extras
  imports
    HOL-Probability.Probability
    PiE-Rel-Extras
begin

context prob-space

```

**begin**

**lemma** *prob-sum-Union*:

**assumes** *measurable*: *finite A*  $A \subseteq \text{events}$  *disjoint A*

**shows**  $\text{prob} (\bigcup A) = (\sum_{e \in A} \text{prob} (e))$

*<proof>*

**lemma** *events-inter*:

**assumes** *finite S*

**assumes**  $S \neq \{\}$

**shows**  $(\bigwedge A. A \in S \implies A \in \text{events}) \implies \bigcap S \in \text{events}$

*<proof>*

**lemma** *events-union*:

**assumes** *finite S*

**shows**  $(\bigwedge A. A \in S \implies A \in \text{events}) \implies \bigcup S \in \text{events}$

*<proof>*

**lemma** *prob-inter-set-lt-elem*:  $A \in \text{events} \implies \text{prob} (A \cap (\bigcap AS)) \leq \text{prob} A$

*<proof>*

**lemma** *Inter-event-ss*:  $\text{finite } A \implies A \subseteq \text{events} \implies A \neq \{\} \implies \bigcap A \in \text{events}$

*<proof>*

**lemma** *prob-inter-ss-lt*:

**assumes** *finite A*

**assumes**  $A \subseteq \text{events}$

**assumes**  $B \neq \{\}$

**assumes**  $B \subseteq A$

**shows**  $\text{prob} (\bigcap A) \leq \text{prob} (\bigcap B)$

*<proof>*

**lemma** *prob-inter-ss-lt-index*:

**assumes** *finite A*

**assumes**  $F \text{ ' } A \subseteq \text{events}$

**assumes**  $B \neq \{\}$

**assumes**  $B \subseteq A$

**shows**  $\text{prob} (\bigcap (F \text{ ' } A)) \leq \text{prob} (\bigcap (F \text{ ' } B))$

*<proof>*

**lemma** *space-compl-double*:

**assumes**  $S \subseteq \text{events}$

**shows**  $((-) (\text{space } M)) \text{ ' } (((-) (\text{space } M)) \text{ ' } S) = S$

*<proof>*

**lemma** *bij-betw-compl-sets*:

**assumes**  $S \subseteq \text{events}$

**assumes**  $S' = ((-) (\text{space } M)) \text{ ' } S$

**shows** *bij-betw*  $((-) (\text{space } M)) S' S$

*<proof>*

**lemma** *bij-betw-compl-sets-rev*:

**assumes**  $S \subseteq \text{events}$

**assumes**  $S' = ((-) (\text{space } M)) ' S$

**shows** *bij-betw*  $((-) (\text{space } M)) S S'$

*<proof>*

**lemma** *prob0-basic-inter*:  $A \in \text{events} \implies B \in \text{events} \implies \text{prob } A = 0 \implies \text{prob } (A \cap B) = 0$

*<proof>*

**lemma** *prob0-basic-Inter*:  $A \in \text{events} \implies B \subseteq \text{events} \implies \text{prob } A = 0 \implies \text{prob } (A \cap (\bigcap B)) = 0$

*<proof>*

**lemma** *prob1-basic-inter*:  $A \in \text{events} \implies B \in \text{events} \implies \text{prob } A = 1 \implies \text{prob } (A \cap B) = \text{prob } B$

*<proof>*

**lemma** *prob1-basic-Inter*:

**assumes**  $A \in \text{events}$   $B \subseteq \text{events}$

**assumes**  $\text{prob } A = 1$

**assumes**  $B \neq \{\}$

**assumes** *finite*  $B$

**shows**  $\text{prob } (A \cap (\bigcap B)) = \text{prob } (\bigcap B)$

*<proof>*

**lemma** *compl-identity*:  $A \in \text{events} \implies \text{space } M - (\text{space } M - A) = A$

*<proof>*

**lemma** *prob-addition-rule*:  $A \in \text{events} \implies B \in \text{events} \implies$

$\text{prob } (A \cup B) = \text{prob } A + \text{prob } B - \text{prob } (A \cap B)$

*<proof>*

**lemma** *compl-subset-in-events*:  $S \subseteq \text{events} \implies (-) (\text{space } M) ' S \subseteq \text{events}$

*<proof>*

**lemma** *prob-compl-diff-inter*:  $A \in \text{events} \implies B \in \text{events} \implies$

$\text{prob } (A \cap (\text{space } M - B)) = \text{prob } A - \text{prob } (A \cap B)$

*<proof>*

**lemma** *bij-betw-prod-prob*: *bij-betw*  $f A B \implies (\prod_{b \in B}. \text{prob } b) = (\prod_{a \in A}. \text{prob } (f a))$

*<proof>*

**definition** *event-compl* :: 'a set  $\Rightarrow$  'a set **where**

*event-compl*  $A \equiv \text{space } M - A$

**lemma compl-Union:**  $A \neq \{\}$   $\implies$   $\text{space } M - (\bigcup A) = (\bigcap a \in A . (\text{space } M - a))$   
 ⟨proof⟩

**lemma compl-Union-fn:**  $A \neq \{\}$   $\implies$   $\text{space } M - (\bigcup (F \cdot A)) = (\bigcap a \in A . (\text{space } M - F a))$   
 ⟨proof⟩

**end**

Reasoning on the probability of function sets

**lemma card-PiE-val-ss-eq:**

**assumes** *finite*  $A$   
**assumes**  $b \in B$   
**assumes**  $d \subseteq A$   
**assumes**  $B \neq \{\}$   
**assumes** *finite*  $B$   
**shows**  $\text{card } \{f \in (A \rightarrow_E B) . (\forall v \in d . f v = b)\} / \text{card } (A \rightarrow_E B) = 1 / ((\text{card } B) \text{ powi } (\text{card } d))$   
**(is**  $\text{card } \{f \in ?C . (\forall v \in d . f v = b)\} / \text{card } ?C = 1 / ((\text{card } B) \text{ powi } (\text{card } d))$   
 ⟨proof⟩

**lemma card-PiE-val-indiv-eq:**

**assumes** *finite*  $A$   
**assumes**  $b \in B$   
**assumes**  $d \in A$   
**assumes**  $B \neq \{\}$   
**assumes** *finite*  $B$   
**shows**  $\text{card } \{f \in (A \rightarrow_E B) . f d = b\} / \text{card } (A \rightarrow_E B) = 1 / (\text{card } B)$   
**(is**  $\text{card } \{f \in ?C . f d = b\} / \text{card } ?C = 1 / (\text{card } B)$   
 ⟨proof⟩

**lemma prob-uniform-ex-fun-space:**

**assumes** *finite*  $A$   
**assumes**  $b \in B$   
**assumes**  $d \subseteq A$   
**assumes**  $B \neq \{\}$   
**assumes**  $A \neq \{\}$   
**assumes** *finite*  $B$   
**shows**  $\text{prob-space.prob } (\text{uniform-count-measure } (A \rightarrow_E B)) \{f \in (A \rightarrow_E B) . (\forall v \in d . f v = b)\} =$   
 $1 / ((\text{card } B) \text{ powi } (\text{card } d))$   
 ⟨proof⟩

**proposition integrable-uniform-count-measure-finite:**

**fixes**  $g :: 'a \Rightarrow 'b :: \{\text{banach, second-countable-topology}\}$   
**shows** *finite*  $A \implies \text{integrable } (\text{uniform-count-measure } A) g$   
 ⟨proof⟩

**end**



## 4 Conditional Probability Library Extensions

```
theory Cond-Prob-Extensions
  imports
    Prob-Events-Extras
    Design-Theory.Multisets-Extras
begin
```

### 4.1 Miscellaneous Set and List Lemmas

```
lemma nth-image-tl:
  assumes  $xs \neq []$ 
  shows  $nth\ xs\ ' \{1..<length\ xs\} = set(tl\ xs)$ 
<proof>
```

```
lemma exists-list-card:
  assumes finite S
  obtains xs where  $set\ xs = S$  and  $length\ xs = card\ S$ 
<proof>
```

```
lemma bij-betw-inter-empty:
  assumes bij-betw f A B
  assumes  $A' \subseteq A$ 
  assumes  $A'' \subseteq A$ 
  assumes  $A' \cap A'' = \{\}$ 
  shows  $f\ ' \ A' \cap f\ ' \ A'' = \{\}$ 
<proof>
```

```
lemma bij-betw-image-comp-eq:
  assumes bij-betw g T S
  shows  $(F \circ g)\ ' \ T = F\ ' \ S$ 
<proof>
```

```
lemma prod-card-image-set-eq:
  assumes bij-betw f  $\{0..<card\ S\}$  S
  assumes finite S
  shows  $(\prod i \in \{n..<(card\ S)\} . g\ (f\ i)) = (\prod i \in f\ ' \ \{n..<card\ S\} . g\ i)$ 
<proof>
```

```
lemma set-take-distinct-elem-not:
  assumes distinct xs
  assumes  $i < length\ xs$ 
  shows  $xs\ !\ i \notin set\ (take\ i\ xs)$ 
<proof>
```

### 4.2 Conditional Probability Basics

```
context prob-space
begin
```

Abbreviation to mirror mathematical notations

**abbreviation** *cond-prob-ev* :: 'a set  $\Rightarrow$  'a set  $\Rightarrow$  real ( $\mathcal{P}'(- | -')$ ) **where**  
 $\mathcal{P}(B | A) \equiv \mathcal{P}(x \text{ in } M. (x \in B) | (x \in A))$

**lemma** *cond-prob-inter*:  $\mathcal{P}(B | A) = \mathcal{P}(\omega \text{ in } M. (\omega \in B \cap A)) / \mathcal{P}(\omega \text{ in } M. (\omega \in A))$   
*<proof>*

**lemma** *cond-prob-ev-def*:  
**assumes**  $A \in \text{events } B \in \text{events}$   
**shows**  $\mathcal{P}(B | A) = \text{prob } (A \cap B) / \text{prob } A$   
*<proof>*

**lemma** *measurable-in-ev*:  
**assumes**  $A \in \text{events}$   
**shows**  $\text{Measurable.pred } M (\lambda x . x \in A)$   
*<proof>*

**lemma** *measure-uniform-measure-eq-cond-prob-ev*:  
**assumes**  $A \in \text{events } B \in \text{events}$   
**shows**  $\mathcal{P}(A | B) = \mathcal{P}(x \text{ in uniform-measure } M \{x \in \text{space } M. x \in B\}. x \in A)$   
*<proof>*

**lemma** *measure-uniform-measure-eq-cond-prob-ev2*:  
**assumes**  $A \in \text{events } B \in \text{events}$   
**shows**  $\mathcal{P}(A | B) = \text{measure } (\text{uniform-measure } M \{x \in \text{space } M. x \in B\}) A$   
*<proof>*

**lemma** *measure-uniform-measure-eq-cond-prob-ev3*:  
**assumes**  $A \in \text{events } B \in \text{events}$   
**shows**  $\mathcal{P}(A | B) = \text{measure } (\text{uniform-measure } M B) A$   
*<proof>*

**lemma** *prob-space-cond-prob-uniform*:  
**assumes**  $\text{prob } (\{x \in \text{space } M. Q x\}) > 0$   
**shows**  $\text{prob-space } (\text{uniform-measure } M \{x \in \text{space } M. Q x\})$   
*<proof>*

**lemma** *prob-space-cond-prob-event*:  
**assumes**  $\text{prob } B > 0$   
**shows**  $\text{prob-space } (\text{uniform-measure } M B)$   
*<proof>*

Note this case shouldn't be used. Conditional probability should have  $> 0$  assumption

**lemma** *cond-prob-empty*:  $\mathcal{P}(B | \{\}) = 0$   
*<proof>*

**lemma** *cond-prob-space*:  $\mathcal{P}(A | \text{space } M) = \mathcal{P}(w \text{ in } M . w \in A)$

*<proof>*

**lemma** *cond-prob-space-ev*: **assumes**  $A \in \text{events}$  **shows**  $\mathcal{P}(A \mid \text{space } M) = \text{prob } A$

*<proof>*

**lemma** *cond-prob-UNIV*:  $\mathcal{P}(A \mid \text{UNIV}) = \mathcal{P}(w \text{ in } M . w \in A)$

*<proof>*

**lemma** *cond-prob-UNIV-ev*:  $A \in \text{events} \implies \mathcal{P}(A \mid \text{UNIV}) = \text{prob } A$

*<proof>*

**lemma** *cond-prob-neg*:

**assumes**  $A \in \text{events}$   $B \in \text{events}$

**assumes**  $\text{prob } A > 0$

**shows**  $\mathcal{P}(\text{space } M - B \mid A) = 1 - \mathcal{P}(B \mid A)$

*<proof>*

### 4.3 Bayes Theorem

**lemma** *prob-intersect-A*:

**assumes**  $A \in \text{events}$   $B \in \text{events}$

**shows**  $\text{prob } (A \cap B) = \text{prob } A * \mathcal{P}(B \mid A)$

*<proof>*

**lemma** *prob-intersect-B*:

**assumes**  $A \in \text{events}$   $B \in \text{events}$

**shows**  $\text{prob } (A \cap B) = \text{prob } B * \mathcal{P}(A \mid B)$

*<proof>*

**theorem** *Bayes-theorem*:

**assumes**  $A \in \text{events}$   $B \in \text{events}$

**shows**  $\text{prob } B * \mathcal{P}(A \mid B) = \text{prob } A * \mathcal{P}(B \mid A)$

*<proof>*

**corollary** *Bayes-theorem-div*:

**assumes**  $A \in \text{events}$   $B \in \text{events}$

**shows**  $\mathcal{P}(A \mid B) = (\text{prob } A * \mathcal{P}(B \mid A)) / (\text{prob } B)$

*<proof>*

**lemma** *cond-prob-dual-intersect*:

**assumes**  $A \in \text{events}$   $B \in \text{events}$   $C \in \text{events}$

**assumes**  $\text{prob } C \neq 0$

**shows**  $\mathcal{P}(A \mid (B \cap C)) = \mathcal{P}(A \cap B \mid C) / \mathcal{P}(B \mid C)$  (is ?LHS = ?RHS)

*<proof>*

**lemma** *cond-prob-ev-double*:

**assumes**  $A \in \text{events}$   $B \in \text{events}$   $C \in \text{events}$

**assumes**  $\text{prob } C > 0$   
**shows**  $\mathcal{P}(x \text{ in } (\text{uniform-measure } M \ C). (x \in A) \mid (x \in B)) = \mathcal{P}(A \mid (B \cap C))$   
 <proof>

**lemma** *cond-prob-inter-set-lt*:  
**assumes**  $A \in \text{events } B \in \text{events } AS \subseteq \text{events}$   
**assumes** *finite AS*  
**shows**  $\mathcal{P}((A \cap (\bigcap AS)) \mid B) \leq \mathcal{P}(A \mid B)$  (is ?LHS  $\leq$  ?RHS)  
 <proof>

#### 4.4 Conditional Probability Multiplication Rule

Many list and indexed variations of this lemma

**lemma** *prob-cond-Inter-List*:  
**assumes**  $xs \neq []$   
**assumes**  $\bigwedge A. A \in \text{set } xs \implies A \in \text{events}$   
**shows**  $\text{prob } (\bigcap (\text{set } xs)) = \text{prob } (\text{hd } xs) * (\prod i = 1..<(\text{length } xs) . \mathcal{P}((xs ! i) \mid (\bigcap (\text{set } (\text{take } i \ xs))))))$   
 <proof>

**lemma** *prob-cond-Inter-index*:  
**fixes**  $n :: \text{nat}$   
**assumes**  $n > 0$   
**assumes**  $F \text{ ' } \{0..<n\} \subseteq \text{events}$   
**shows**  $\text{prob } (\bigcap (F \text{ ' } \{0..<n\} )) = \text{prob } (F \ 0) * (\prod i \in \{1..<n\} . \mathcal{P}(F \ i \mid (\bigcap (F \text{ ' } \{0..<i\}))))$   
 <proof>

**lemma** *prob-cond-Inter-index-compl*:  
**fixes**  $n :: \text{nat}$   
**assumes**  $n > 0$   
**assumes**  $F \text{ ' } \{0..<n\} \subseteq \text{events}$   
**shows**  $\text{prob } (\bigcap x \in \{0..<n\} . \text{space } M - F \ x) = \text{prob } (\text{space } M - F \ 0) * (\prod i \in \{1..<n\} . \mathcal{P}(\text{space } M - F \ i \mid (\bigcap j \in \{0..<i\} . \text{space } M - F \ j)))$   
 <proof>

**lemma** *prob-cond-Inter-take-cond*:  
**assumes**  $xs \neq []$   
**assumes**  $\text{set } xs \subseteq \text{events}$   
**assumes**  $S \subseteq \text{events}$   
**assumes**  $S \neq \{\}$   
**assumes** *finite S*  
**assumes**  $\text{prob } (\bigcap S) > 0$   
**shows**  $\mathcal{P}((\bigcap (\text{set } xs)) \mid (\bigcap S)) = (\prod i = 0..<(\text{length } xs) . \mathcal{P}((xs ! i) \mid (\bigcap (\text{set } (\text{take } i \ xs) \cup S))))$   
 <proof>

**lemma** *prob-cond-Inter-index-cond-set:*

**fixes**  $n :: nat$   
**assumes**  $n > 0$   
**assumes** *finite*  $E$   
**assumes**  $E \neq \{\}$   
**assumes**  $E \subseteq events$   
**assumes**  $F \text{ ' } \{0..<n\} \subseteq events$   
**assumes**  $prob (\bigcap E) > 0$   
**shows**  $\mathcal{P}((\bigcap (F \text{ ' } \{0..<n\})) \mid (\bigcap E)) = (\prod i \in \{0..<n\}. \mathcal{P}(F i \mid (\bigcap ((F \text{ ' } \{0..<i\}) \cup E))))$   
*<proof>*

**lemma** *prob-cond-Inter-index-cond-compl-set:*

**fixes**  $n :: nat$   
**assumes**  $n > 0$   
**assumes** *finite*  $E$   
**assumes**  $E \neq \{\}$   
**assumes**  $E \subseteq events$   
**assumes**  $F \text{ ' } \{0..<n\} \subseteq events$   
**assumes**  $prob (\bigcap E) > 0$   
**shows**  $\mathcal{P}((\bigcap ((-) (space M) \text{ ' } F \text{ ' } \{0..<n\})) \mid (\bigcap E)) = (\prod i = 0..<n. \mathcal{P}((space M - F i) \mid (\bigcap ((-) (space M) \text{ ' } F \text{ ' } \{0..<i\} \cup E))))$   
*<proof>*

**lemma** *prob-cond-Inter-index-cond:*

**fixes**  $n :: nat$   
**assumes**  $n > 0$   
**assumes**  $n < m$   
**assumes**  $F \text{ ' } \{0..<m\} \subseteq events$   
**assumes**  $prob (\bigcap j \in \{n..<m\}. F j) > 0$   
**shows**  $\mathcal{P}((\bigcap (F \text{ ' } \{0..<n\})) \mid (\bigcap j \in \{n..<m\}. F j)) = (\prod i \in \{0..<n\}. \mathcal{P}(F i \mid (\bigcap ((F \text{ ' } \{0..<i\}) \cup (F \text{ ' } \{n..<m\}))))$   
*<proof>*

**lemma** *prob-cond-Inter-index-cond-compl:*

**fixes**  $n :: nat$   
**assumes**  $n > 0$   
**assumes**  $n < m$   
**assumes**  $F \text{ ' } \{0..<m\} \subseteq events$   
**assumes**  $prob (\bigcap j \in \{n..<m\}. F j) > 0$   
**shows**  $\mathcal{P}((\bigcap ((-) (space M) \text{ ' } F \text{ ' } \{0..<n\})) \mid (\bigcap (F \text{ ' } \{n..<m\}))) = (\prod i = 0..<n. \mathcal{P}((space M - F i) \mid (\bigcap ((-) (space M) \text{ ' } F \text{ ' } \{0..<i\} \cup (F \text{ ' } \{n..<m\}))))$   
*<proof>*

**lemma** *prob-cond-Inter-take-cond-neg:*

**assumes**  $xs \neq []$   
**assumes**  $set xs \subseteq events$

**assumes**  $S \subseteq \text{events}$   
**assumes**  $S \neq \{\}$   
**assumes** *finite*  $S$   
**assumes**  $\text{prob} (\bigcap S) > 0$   
**shows**  $\mathcal{P}((\bigcap ((-) (\text{space } M) \text{ ' } (\text{set } xs))) \mid (\bigcap S)) =$   
 $(\prod_{i = 0..<(\text{length } xs)} . \mathcal{P}((\text{space } M - xs ! i) \mid (\bigcap ((-) (\text{space } M) \text{ ' } (\text{set } (\text{take } i \text{ } xs)) \cup S))))$   
 $\langle \text{proof} \rangle$

**lemma** *prob-cond-Inter-List-Index:*

**assumes**  $xs \neq []$   
**assumes**  $\text{set } xs \subseteq \text{events}$   
**shows**  $\text{prob} (\bigcap (\text{set } xs)) = \text{prob} (\text{hd } xs) * (\prod_{i = 1..<(\text{length } xs)} .$   
 $\mathcal{P}((xs ! i) \mid (\bigcap_{j \in \{0..<i\}} . xs ! j)))$   
 $\langle \text{proof} \rangle$

**lemma** *obtains-prob-cond-Inter-index:*

**assumes**  $S \neq \{\}$   
**assumes**  $S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**obtains**  $xs$  **where**  $\text{set } xs = S$  **and**  $\text{length } xs = \text{card } S$  **and**  
 $\text{prob} (\bigcap S) = \text{prob} (\text{hd } xs) * (\prod_{i = 1..<(\text{length } xs)} . \mathcal{P}((xs ! i) \mid (\bigcap_{j \in \{0..<i\}} . xs ! j)))$   
 $\langle \text{proof} \rangle$

**lemma** *obtain-list-index:*

**assumes** *bij-betw*  $g \{0..<\text{card } S\} S$   
**assumes** *finite*  $S$   
**obtains**  $xs$  **where**  $\text{set } xs = S$  **and**  $\bigwedge i . i \in \{0..<\text{card } S\} \implies g \ i = xs ! i$  **and**  
*distinct*  $xs$   
 $\langle \text{proof} \rangle$

**lemma** *prob-cond-inter-fn:*

**assumes** *bij-betw*  $g \{0..<\text{card } S\} S$   
**assumes** *finite*  $S$   
**assumes**  $S \neq \{\}$   
**assumes**  $S \subseteq \text{events}$   
**shows**  $\text{prob} (\bigcap S) = \text{prob} (g \ 0) * (\prod_{i \in \{1..<(\text{card } S)\}} . \mathcal{P}(g \ i \mid (\bigcap (g \text{ ' } \{0..<i\}))))$   
 $\langle \text{proof} \rangle$

**lemma** *prob-cond-inter-obtain-fn:*

**assumes**  $S \neq \{\}$   
**assumes**  $S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**obtains**  $f$  **where** *bij-betw*  $f \{0..<\text{card } S\} S$  **and**  
 $\text{prob} (\bigcap S) = \text{prob} (f \ 0) * (\prod_{i \in \{1..<(\text{card } S)\}} . \mathcal{P}(f \ i \mid (\bigcap (f \text{ ' } \{0..<i\}))))$   
 $\langle \text{proof} \rangle$

**lemma** *prob-cond-inter-obtain-fn-compl:*

**assumes**  $S \neq \{\}$   
**assumes**  $S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**obtains**  $f$  **where** *bij-betw*  $f \{0..<\text{card } S\} S$  **and**  $\text{prob} (\bigcap ((-) (\text{space } M) ' S))$   
 $=$   
 $\text{prob} (\text{space } M - f 0) * (\prod i \in \{1..<(\text{card } S)\} . \mathcal{P}(\text{space } M - f i \mid (\bigcap ((-) (\text{space } M) ' f ' \{0..<i\}))))$   
*<proof>*

**lemma** *prob-cond-Inter-index-cond-fn:*

**assumes**  $I \neq \{\}$   
**assumes** *finite*  $I$   
**assumes** *finite*  $E$   
**assumes**  $E \neq \{\}$   
**assumes**  $E \subseteq \text{events}$   
**assumes**  $F ' I \subseteq \text{events}$   
**assumes**  $\text{prob} (\bigcap E) > 0$   
**assumes** *bb: bij-betw*  $g \{0..<\text{card } I\} I$   
**shows**  $\mathcal{P}((\bigcap (F ' g ' \{0..<\text{card } I\})) \mid (\bigcap E)) =$   
 $(\prod i \in \{0..<\text{card } I\} . \mathcal{P}(F (g i) \mid (\bigcap ((F ' g ' \{0..<i\}) \cup E))))$   
*<proof>*

**lemma** *prob-cond-Inter-index-cond-obtains:*

**assumes**  $I \neq \{\}$   
**assumes** *finite*  $I$   
**assumes** *finite*  $E$   
**assumes**  $E \neq \{\}$   
**assumes**  $E \subseteq \text{events}$   
**assumes**  $F ' I \subseteq \text{events}$   
**assumes**  $\text{prob} (\bigcap E) > 0$   
**obtains**  $g$  **where** *bij-betw*  $g \{0..<\text{card } I\} I$  **and**  $\mathcal{P}((\bigcap (F ' g ' \{0..<\text{card } I\})) \mid (\bigcap E)) =$   
 $(\prod i \in \{0..<\text{card } I\} . \mathcal{P}(F (g i) \mid (\bigcap ((F ' g ' \{0..<i\}) \cup E))))$   
*<proof>*

**lemma** *prob-cond-Inter-index-cond-compl-fn:*

**assumes**  $I \neq \{\}$   
**assumes** *finite*  $I$   
**assumes** *finite*  $E$   
**assumes**  $E \neq \{\}$   
**assumes**  $E \subseteq \text{events}$   
**assumes**  $F ' I \subseteq \text{events}$   
**assumes**  $\text{prob} (\bigcap E) > 0$   
**assumes** *bb: bij-betw*  $g \{0..<\text{card } I\} I$   
**shows**  $\mathcal{P}((\bigcap Aj \in I . \text{space } M - F Aj) \mid (\bigcap E)) =$   
 $(\prod i \in \{0..<\text{card } I\} . \mathcal{P}(\text{space } M - F (g i) \mid (\bigcap (((\lambda Aj . \text{space } M - F Aj) ' g ' \{0..<i\}) \cup E))))$   
*<proof>*

**lemma** *prob-cond-Inter-index-cond-compl-obtains:*

**assumes**  $I \neq \{\}$   
**assumes** *finite*  $I$   
**assumes** *finite*  $E$   
**assumes**  $E \neq \{\}$   
**assumes**  $E \subseteq \text{events}$   
**assumes**  $F \text{ ' } I \subseteq \text{events}$   
**assumes**  $\text{prob} (\bigcap E) > 0$   
**obtains**  $g$  **where** *bij-betw*  $g \{0..<\text{card } I\} I$  **and**  $\mathcal{P}((\bigcap Aj \in I . \text{space } M - F Aj)$   
 $| (\bigcap E)) =$   
 $(\prod i \in \{0..<\text{card } I\} . \mathcal{P}(\text{space } M - F (g i) | (\bigcap ((\lambda Aj . \text{space } M - F Aj) \text{ ' } g \text{ ' }$   
 $\{0..<i\} \cup E))))$   
*<proof>*

**lemma** *prob-cond-inter-index-fn2:*

**assumes**  $F \text{ ' } S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**assumes**  $\text{card } S > 0$   
**assumes** *bij-betw*  $g \{0..<\text{card } S\} S$   
**shows**  $\text{prob} (\bigcap (F \text{ ' } S)) = \text{prob} (F (g 0)) * (\prod i \in \{1..<(\text{card } S)\} . \mathcal{P}(F (g i) |$   
 $(\bigcap (F \text{ ' } g \text{ ' } \{0..<i\}))))$   
*<proof>*

**lemma** *prob-cond-inter-index-fn:*

**assumes**  $F \text{ ' } S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**assumes**  $S \neq \{\}$   
**assumes** *bij-betw*  $g \{0..<\text{card } S\} S$   
**shows**  $\text{prob} (\bigcap (F \text{ ' } S)) = \text{prob} (F (g 0)) * (\prod i \in \{1..<(\text{card } S)\} . \mathcal{P}(F (g i) |$   
 $(\bigcap (F \text{ ' } g \text{ ' } \{0..<i\}))))$   
*<proof>*

**lemma** *prob-cond-inter-index-obtain-fn:*

**assumes**  $F \text{ ' } S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**assumes**  $S \neq \{\}$   
**obtains**  $g$  **where** *bij-betw*  $g \{0..<\text{card } S\} S$  **and**  
 $\text{prob} (\bigcap (F \text{ ' } S)) = \text{prob} (F (g 0)) * (\prod i \in \{1..<(\text{card } S)\} . \mathcal{P}(F (g i) | (\bigcap (F \text{ ' }$   
 $g \text{ ' } \{0..<i\}))))$   
*<proof>*

**lemma** *prob-cond-inter-index-fn-compl:*

**assumes**  $S \neq \{\}$   
**assumes**  $F \text{ ' } S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**assumes** *bij-betw*  $f \{0..<\text{card } S\} S$   
**shows**  $\text{prob} (\bigcap ((-) (\text{space } M) \text{ ' } F \text{ ' } S)) = \text{prob} (\text{space } M - F (f 0)) *$   
 $(\prod i \in \{1..<(\text{card } S)\} . \mathcal{P}(\text{space } M - F (f i) | (\bigcap ((-) (\text{space } M) \text{ ' } f \text{ ' }$



{0..<i})))  
 <proof>

**lemma** *prob-cond-inter-index-obtain-fn-compl*:

**assumes**  $S \neq \{\}$   
**assumes**  $F \text{ ' } S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**obtains**  $f$  **where** *bij-betw*  $f \{0..<card\ S\} S$  **and**  
 $prob (\bigcap ((-) (space\ M) \text{ ' } F \text{ ' } S)) = prob (space\ M - F (f\ 0)) *$   
 $(\prod i \in \{1..<(card\ S)\} . \mathcal{P}(space\ M - F (f\ i) \mid (\bigcap ((-) (space\ M) \text{ ' } F \text{ ' } f \text{ ' } \{0..<i\}))))$   
 <proof>

**lemma** *prob-cond-Inter-take*:

**assumes**  $S \neq \{\}$   
**assumes**  $S \subseteq \text{events}$   
**assumes** *finite*  $S$   
**obtains**  $xs$  **where** *set*  $xs = S$  **and** *length*  $xs = card\ S$  **and**  
 $prob (\bigcap S) = prob (hd\ xs) * (\prod i = 1..<(length\ xs) . \mathcal{P}((xs\ !\ i) \mid (\bigcap (set\ (take\ i\ xs))))))$   
 <proof>

**lemma** *prob-cond-Inter-set-bound*:

**assumes**  $A \neq \{\}$   
**assumes**  $A \subseteq \text{events}$   
**assumes** *finite*  $A$   
**assumes**  $\bigwedge Ai . f\ Ai \geq 0 \wedge f\ Ai \leq 1$   
**assumes**  $\bigwedge Ai\ S. Ai \in A \implies S \subseteq A - \{Ai\} \implies S \neq \{\} \implies \mathcal{P}(Ai \mid (\bigcap S)) \geq f\ Ai$   
**assumes**  $\bigwedge Ai. Ai \in A \implies prob\ Ai \geq f\ Ai$   
**shows**  $prob (\bigcap A) \geq (\prod a' \in A . f\ a')$   
 <proof>  
 end

end

## 5 Independent Events

**theory** *Indep-Events* **imports** *Cond-Prob-Extensions*  
**begin**

### 5.1 More bijection helpers

**lemma** *bij-betw-obtain-subsetr*:

**assumes** *bij-betw*  $f\ A\ B$   
**assumes**  $A' \subseteq A$   
**obtains**  $B'$  **where**  $B' \subseteq B$  **and**  $B' = f \text{ ' } A'$

*<proof>*

**lemma** *bij-betw-obtain-subset*:

**assumes** *bij-betw*  $f A B$

**assumes**  $B' \subseteq B$

**obtains**  $A'$  **where**  $A' \subseteq A$  **and**  $B' = f ' A'$

*<proof>*

**lemma** *bij-betw-remove*:  $bij-betw f A B \implies a \in A \implies bij-betw f (A - \{a\}) (B - \{f a\})$

*<proof>*

## 5.2 Independent Event Extensions

Extensions on both the *indep\_event* definition and the *indep\_events* definition

**context** *prob-space*

**begin**

**lemma** *indep-eventsD*:  $indep-events A I \implies (A'I \subseteq events) \implies J \subseteq I \implies J \neq \{\} \implies finite J \implies$

$prob (\bigcap_{j \in J}. A j) = (\prod_{j \in J}. prob (A j))$

*<proof>*

**lemma**

**assumes** *indep*: *indep-event*  $A B$

**shows** *indep-eventD-ev1*:  $A \in events$

**and** *indep-eventD-ev2*:  $B \in events$

*<proof>*

**lemma** *indep-eventD*:

**assumes** *ie*: *indep-event*  $A B$

**shows**  $prob (A \cap B) = prob (A) * prob (B)$

*<proof>*

**lemma** *indep-eventI[intro]*:

**assumes** *ev*:  $A \in events B \in events$

**and** *indep*:  $prob (A \cap B) = prob A * prob B$

**shows** *indep-event*  $A B$

*<proof>*

Alternate set definition - when no possibility of duplicate objects

**definition** *indep-events-set* :: 'a set  $\Rightarrow bool$  **where**

*indep-events-set*  $E \equiv (E \subseteq events \wedge (\forall J. J \subseteq E \longrightarrow finite J \longrightarrow J \neq \{\} \longrightarrow prob (\bigcap J) = (\prod_{i \in J}. prob i)))$

**lemma** *indep-events-setI[intro]*:  $E \subseteq events \implies (\bigwedge J. J \subseteq E \implies finite J \implies J \neq \{\} \implies$

$prob (\bigcap J) = (\prod_{i \in J}. prob i) \implies indep-events-set E$

*<proof>*

**lemma** *indep-events-subset:*

*indep-events-set*  $E \longleftrightarrow (\forall J \subseteq E. \text{indep-events-set } J)$

*<proof>*

**lemma** *indep-events-subset2:*

*indep-events-set*  $E \implies J \subseteq E \implies \text{indep-events-set } J$

*<proof>*

**lemma** *indep-events-set-events:* *indep-events-set*  $E \implies (\bigwedge e. e \in E \implies e \in \text{events})$

*<proof>*

**lemma** *indep-events-set-events-ss:* *indep-events-set*  $E \implies E \subseteq \text{events}$

*<proof>*

**lemma** *indep-events-set-probs:* *indep-events-set*  $E \implies J \subseteq E \implies \text{finite } J \implies J \neq \{\}$

$\text{prob } (\bigcap J) = (\prod_{i \in J} \text{prob } i)$

*<proof>*

**lemma** *indep-events-set-prod-all:* *indep-events-set*  $E \implies \text{finite } E \implies E \neq \{\} \implies$

$\text{prob } (\bigcap E) = \text{prod prob } E$

*<proof>*

**lemma** *indep-events-not-contain-compl:*

**assumes** *indep-events-set*  $E$

**assumes**  $A \in E$

**assumes**  $\text{prob } A > 0 \text{ prob } A < 1$

**shows**  $(\text{space } M - A) \notin E$  (**is**  $?A' \notin E$ )

*<proof>*

**lemma** *indep-events-contain-compl-prob01:*

**assumes** *indep-events-set*  $E$

**assumes**  $A \in E$

**assumes**  $\text{space } M - A \in E$

**shows**  $\text{prob } A = 0 \vee \text{prob } A = 1$

*<proof>*

**lemma** *indep-events-set-singleton:*

**assumes**  $A \in \text{events}$

**shows** *indep-events-set*  $\{A\}$

*<proof>*

**lemma** *indep-events-pairs:*

**assumes** *indep-events-set*  $S$

**assumes**  $A \in S \ B \in S \ A \neq B$

**shows** *indep-event*  $A B$   
*<proof>*

**lemma** *indep-events-inter-pairs*:  
**assumes** *indep-events-set*  $S$   
**assumes** *finite*  $A$  *finite*  $B$   
**assumes**  $A \neq \{\}$   $B \neq \{\}$   
**assumes**  $A \subseteq S$   $B \subseteq S$   $A \cap B = \{\}$   
**shows** *indep-event*  $(\bigcap A)$   $(\bigcap B)$   
*<proof>*

**lemma** *indep-events-inter-single*:  
**assumes** *indep-events-set*  $S$   
**assumes** *finite*  $B$   
**assumes**  $B \neq \{\}$   
**assumes**  $A \in S$   $B \subseteq S$   $A \notin B$   
**shows** *indep-event*  $A$   $(\bigcap B)$   
*<proof>*

**lemma** *indep-events-set-prob1*:  
**assumes**  $A \in \text{events}$   
**assumes** *prob*  $A = 1$   
**assumes**  $A \notin S$   
**assumes** *indep-events-set*  $S$   
**shows** *indep-events-set*  $(S \cup \{A\})$   
*<proof>*

**lemma** *indep-events-set-prob0*:  
**assumes**  $A \in \text{events}$   
**assumes** *prob*  $A = 0$   
**assumes**  $A \notin S$   
**assumes** *indep-events-set*  $S$   
**shows** *indep-events-set*  $(S \cup \{A\})$   
*<proof>*

**lemma** *indep-event-commute*:  
**assumes** *indep-event*  $A B$   
**shows** *indep-event*  $B A$   
*<proof>*

Showing complement operation maintains independence

**lemma** *indep-event-one-compl*:  
**assumes** *indep-event*  $A B$   
**shows** *indep-event*  $A$   $(\text{space } M - B)$   
*<proof>*

**lemma** *indep-event-one-compl-rev*:  
**assumes**  $B \in \text{events}$

**assumes** *indep-event*  $A$  (*space*  $M - B$ )  
**shows** *indep-event*  $A$   $B$   
 $\langle$ *proof* $\rangle$

**lemma** *indep-event-double-compl*: *indep-event*  $A$   $B \implies$  *indep-event* (*space*  $M - A$ ) (*space*  $M - B$ )  
 $\langle$ *proof* $\rangle$

**lemma** *indep-event-double-compl-rev*:  $A \in$  *events*  $\implies B \in$  *events*  $\implies$   
*indep-event* (*space*  $M - A$ ) (*space*  $M - B$ )  $\implies$  *indep-event*  $A$   $B$   
 $\langle$ *proof* $\rangle$

**lemma** *indep-events-set-one-compl*:  
**assumes** *indep-events-set*  $S$   
**assumes**  $A \in S$   
**shows** *indep-events-set* ( $\{\text{space } M - A\} \cup (S - \{A\})$ )  
 $\langle$ *proof* $\rangle$

**lemma** *indep-events-set-update-compl*:  
**assumes** *indep-events-set*  $E$   
**assumes**  $E = A \cup B$   
**assumes**  $A \cap B = \{\}$   
**assumes** *finite*  $E$   
**shows** *indep-events-set* ( $((-) \text{ space } M - A) \cup B$ )  
 $\langle$ *proof* $\rangle$

**lemma** *indep-events-set-compl*:  
**assumes** *indep-events-set*  $E$   
**assumes** *finite*  $E$   
**shows** *indep-events-set* ( $(\lambda e. \text{space } M - e) \text{ ' } E$ )  
 $\langle$ *proof* $\rangle$

**lemma** *indep-event-empty*:  
**assumes**  $A \in$  *events*  
**shows** *indep-event*  $A$   $\{\}$   
 $\langle$ *proof* $\rangle$

**lemma** *indep-event-compl-inter*:  
**assumes** *indep-event*  $A$   $C$   
**assumes**  $B \in$  *events*  
**assumes** *indep-event*  $A$  ( $B \cap C$ )  
**shows** *indep-event*  $A$  ( $(\text{space } M - B) \cap C$ )  
 $\langle$ *proof* $\rangle$

**lemma** *indep-events-index-subset*:  
*indep-events*  $F$   $E \longleftrightarrow (\forall J \subseteq E. \text{indep-events } F$   $J)$

*<proof>*

**lemma** *indep-events-index-subset2:*

*indep-events F E  $\implies$  J  $\subseteq$  E  $\implies$  indep-events F J*

*<proof>*

**lemma** *indep-events-events-ss: indep-events F E  $\implies$  F ' E  $\subseteq$  events*

*<proof>*

**lemma** *indep-events-events: indep-events F E  $\implies$  ( $\bigwedge e. e \in E \implies F e \in \text{events}$ )*

*<proof>*

**lemma** *indep-events-probs: indep-events F E  $\implies$  J  $\subseteq$  E  $\implies$  finite J  $\implies$  J  $\neq$  {}  $\implies$  prob ( $\bigcap (F ' J)$ ) = ( $\prod_{i \in J. \text{prob } (F i)}$ )*

*<proof>*

**lemma** *indep-events-prod-all: indep-events F E  $\implies$  finite E  $\implies$  E  $\neq$  {}  $\implies$  prob ( $\bigcap (F ' E)$ ) = ( $\prod_{i \in E. \text{prob } (F i)}$ )*

*<proof>*

**lemma** *indep-events-ev-not-contain-compl:*

**assumes** *indep-events F E*

**assumes** *A  $\in$  E*

**assumes** *prob (F A) > 0 prob (F A) < 1*

**shows** *(space M - F A)  $\notin$  F ' E (is ?A'  $\notin$  F ' E)*

*<proof>*

**lemma** *indep-events-singleton:*

**assumes** *F A  $\in$  events*

**shows** *indep-events F {A}*

*<proof>*

**lemma** *indep-events-ev-pairs:*

**assumes** *indep-events F S*

**assumes** *A  $\in$  S B  $\in$  S A  $\neq$  B*

**shows** *indep-event (F A) (F B)*

*<proof>*

**lemma** *indep-events-ev-inter-pairs:*

**assumes** *indep-events F S*

**assumes** *finite A finite B*

**assumes** *A  $\neq$  {} B  $\neq$  {}*

**assumes** *A  $\subseteq$  S B  $\subseteq$  S A  $\cap$  B = {}*

**shows** *indep-event ( $\bigcap (F ' A)$ ) ( $\bigcap (F ' B)$ )*

*<proof>*

**lemma** *indep-events-ev-inter-single:*

**assumes** *indep-events*  $F S$   
**assumes** *finite*  $B$   
**assumes**  $B \neq \{\}$   
**assumes**  $A \in S \ B \subseteq S \ A \notin B$   
**shows** *indep-event*  $(F A) (\bigcap (F \text{ ` } B))$   
 <proof>

**lemma** *indep-events-fn-eq*:  
**assumes**  $\bigwedge Ai. Ai \in E \implies F Ai = G Ai$   
**assumes** *indep-events*  $F E$   
**shows** *indep-events*  $G E$   
 <proof>

**lemma** *indep-events-fn-eq-iff*:  
**assumes**  $\bigwedge Ai. Ai \in E \implies F Ai = G Ai$   
**shows** *indep-events*  $F E \longleftrightarrow \text{indep-events } G E$   
 <proof>

**lemma** *indep-events-one-compl*:  
**assumes** *indep-events*  $F S$   
**assumes**  $A \in S$   
**shows** *indep-events*  $(\lambda i. \text{if } (i = A) \text{ then } (\text{space } M - F i) \text{ else } F i) S$  (**is**  
*indep-events*  $?G S$ )  
 <proof>

**lemma** *indep-events-update-compl*:  
**assumes** *indep-events*  $F E$   
**assumes**  $E = A \cup B$   
**assumes**  $A \cap B = \{\}$   
**assumes** *finite*  $E$   
**shows** *indep-events*  $(\lambda Ai. \text{if } (Ai \in A) \text{ then } (\text{space } M - (F Ai)) \text{ else } (F Ai)) E$   
 <proof>

**lemma** *indep-events-compl*:  
**assumes** *indep-events*  $F E$   
**assumes** *finite*  $E$   
**shows** *indep-events*  $(\lambda Ai. \text{space } M - F Ai) E$   
 <proof>

**lemma** *indep-events-impl-inj-on*:  
**assumes** *finite*  $A$   
**assumes** *indep-events*  $F A$   
**assumes**  $\bigwedge A'. A' \in A \implies \text{prob } (F A') > 0 \wedge \text{prob } (F A') < 1$   
**shows** *inj-on*  $F A$   
 <proof>

**lemma** *indep-events-imp-set*:  
**assumes** *finite*  $A$   
**assumes** *indep-events*  $F A$

**assumes**  $\bigwedge A' . A' \in A \implies \text{prob } (F A') > 0 \wedge \text{prob } (F A') < 1$   
**shows**  $\text{indep-events-set } (F A)$   
 <proof>

**lemma** *indep-event-set-equiv-bij*:  
**assumes** *bij-betw*  $F A E$   
**assumes** *finite*  $E$   
**shows**  $\text{indep-events-set } E \longleftrightarrow \text{indep-events } F A$   
 <proof>

### 5.3 Mutual Independent Events

Note, set based version only if no duplicates in usage case. The `mutual_indep_events` definition is more general and recommended

**definition** *mutual-indep-set*:: 'a set  $\implies$  'a set set  $\implies$  bool  
**where**  $\text{mutual-indep-set } A S \longleftrightarrow A \in \text{events} \wedge S \subseteq \text{events} \wedge (\forall T \subseteq S . T \neq \{\} \longrightarrow \text{prob } (A \cap (\bigcap T)) = \text{prob } A * \text{prob } (\bigcap T))$

**lemma** *mutual-indep-setI[intro]*:  $A \in \text{events} \implies S \subseteq \text{events} \implies (\bigwedge T . T \subseteq S \implies T \neq \{\}) \implies \text{prob } (A \cap (\bigcap T)) = \text{prob } A * \text{prob } (\bigcap T) \implies \text{mutual-indep-set } A S$   
 <proof>

**lemma** *mutual-indep-setD[dest]*:  $\text{mutual-indep-set } A S \implies T \subseteq S \implies T \neq \{\} \implies \text{prob } (A \cap (\bigcap T)) = \text{prob } A * \text{prob } (\bigcap T)$   
 <proof>

**lemma** *mutual-indep-setD2[dest]*:  $\text{mutual-indep-set } A S \implies A \in \text{events}$   
 <proof>

**lemma** *mutual-indep-setD3[dest]*:  $\text{mutual-indep-set } A S \implies S \subseteq \text{events}$   
 <proof>

**lemma** *mutual-indep-subset*:  $\text{mutual-indep-set } A S \implies T \subseteq S \implies \text{mutual-indep-set } A T$   
 <proof>

**lemma** *mutual-indep-event-set-defD*:  
**assumes**  $\text{mutual-indep-set } A S$   
**assumes** *finite*  $T$   
**assumes**  $T \subseteq S$   
**assumes**  $T \neq \{\}$   
**shows**  $\text{indep-event } A (\bigcap T)$   
 <proof>

**lemma** *mutual-indep-event-defI*:  $A \in \text{events} \implies S \subseteq \text{events} \implies (\bigwedge T . T \subseteq S \implies T \neq \{\}) \implies$



*indep-event*  $A (\bigcap T) \implies$  *mutual-indep-set*  $A S$   
(proof)

**lemma** *mutual-indep-singleton-event*: *mutual-indep-set*  $A S \implies B \in S \implies$  *indep-event*  $A B$   
(proof)

**lemma** *mutual-indep-cond*:  
assumes  $A \in \text{events}$  and  $T \subseteq \text{events}$  and *finite*  $T$   
and *mutual-indep-set*  $A S$  and  $T \subseteq S$  and  $T \neq \{\}$  and  $\text{prob} (\bigcap T) \neq 0$   
shows  $\mathcal{P}(A | (\bigcap T)) = \text{prob } A$   
(proof)

**lemma** *mutual-indep-cond-full*:  
assumes  $A \in \text{events}$  and  $S \subseteq \text{events}$  and *finite*  $S$   
and *mutual-indep-set*  $A S$  and  $S \neq \{\}$  and  $\text{prob} (\bigcap S) \neq 0$   
shows  $\mathcal{P}(A | (\bigcap S)) = \text{prob } A$   
(proof)

**lemma** *mutual-indep-cond-single*:  
assumes  $A \in \text{events}$  and  $B \in \text{events}$   
and *mutual-indep-set*  $A S$  and  $B \in S$  and  $\text{prob } B \neq 0$   
shows  $\mathcal{P}(A | B) = \text{prob } A$   
(proof)

**lemma** *mutual-indep-set-empty*:  $A \in \text{events} \implies$  *mutual-indep-set*  $A \{\}$   
(proof)

**lemma** *not-mutual-indep-set-itself*:  
assumes  $\text{prob } A > 0$  and  $\text{prob } A < 1$   
shows  $\neg$  *mutual-indep-set*  $A \{A\}$   
(proof)

**lemma** *is-mutual-indep-set-itself*:  
assumes  $A \in \text{events}$   
assumes  $\text{prob } A = 0 \vee \text{prob } A = 1$   
shows *mutual-indep-set*  $A \{A\}$   
(proof)

**lemma** *mutual-indep-set-singleton*:  
assumes *indep-event*  $A B$   
shows *mutual-indep-set*  $A \{B\}$   
(proof)

**lemma** *mutual-indep-set-one-compl*:  
assumes *mutual-indep-set*  $A S$   
assumes *finite*  $S$   
assumes  $B \in S$   
shows *mutual-indep-set*  $A (\{\text{space } M - B\} \cup S)$

*<proof>*

**lemma** *mutual-indep-events-set-update-compl:*

**assumes** *mutual-indep-set*  $X E$

**assumes**  $E = A \cup B$

**assumes**  $A \cap B = \{\}$

**assumes** *finite*  $E$

**shows** *mutual-indep-set*  $X ((-) (space M) ' A) \cup B$

*<proof>*

**lemma** *mutual-indep-events-compl:*

**assumes** *finite*  $S$

**assumes** *mutual-indep-set*  $A S$

**shows** *mutual-indep-set*  $A ((\lambda s . space M - s) ' S)$

*<proof>*

**lemma** *mutual-indep-set-all:*

**assumes**  $A \subseteq events$

**assumes**  $\bigwedge Ai. Ai \in A \implies (mutual-indep-set Ai (A - \{Ai\}))$

**shows** *indep-events-set*  $A$

*<proof>*

Prefered version using indexed notation

**definition** *mutual-indep-events:: 'a set  $\implies$  (nat  $\implies$  'a set)  $\implies$  nat set  $\implies$  bool*

**where** *mutual-indep-events*  $A F I \longleftrightarrow A \in events \wedge (F ' I \subseteq events) \wedge (\forall J \subseteq I . J \neq \{\} \implies prob (A \cap (\bigcap j \in J . F j)) = prob A * prob (\bigcap j \in J . F j))$

**lemma** *mutual-indep-eventsI[intro]:*  $A \in events \implies (F ' I \subseteq events) \implies (\bigwedge J. J \subseteq I \implies J \neq \{\}) \implies$

$prob (A \cap (\bigcap j \in J . F j)) = prob A * prob (\bigcap j \in J . F j) \implies mutual-indep-events A F I$

*<proof>*

**lemma** *mutual-indep-eventsD[dest]:*  $mutual-indep-events A F I \implies J \subseteq I \implies J \neq \{\} \implies prob (A \cap (\bigcap j \in J . F j)) = prob A * prob (\bigcap j \in J . F j)$

*<proof>*

**lemma** *mutual-indep-eventsD2[dest]:*  $mutual-indep-events A F I \implies A \in events$

*<proof>*

**lemma** *mutual-indep-eventsD3[dest]:*  $mutual-indep-events A F I \implies F ' I \subseteq events$

*<proof>*

**lemma** *mutual-indep-ev-subset:*  $mutual-indep-events A F I \implies J \subseteq I \implies mutual-indep-events A F J$

*<proof>*

**lemma** *mutual-indep-event-defD*:  
**assumes** *mutual-indep-events A F I*  
**assumes** *finite J*  
**assumes**  $J \subseteq I$   
**assumes**  $J \neq \{\}$   
**shows** *indep-event A*  $(\bigcap j \in J . F j)$   
 $\langle$ *proof* $\rangle$

**lemma** *mutual-ev-indep-event-defI*:  $A \in \text{events} \implies F ' I \subseteq \text{events} \implies (\bigwedge J. J \subseteq I \implies J \neq \{\}) \implies$   
*indep-event A*  $(\bigcap (F ' J)) \implies$  *mutual-indep-events A F I*  
 $\langle$ *proof* $\rangle$

**lemma** *mutual-indep-ev-singleton-event*:  
**assumes** *mutual-indep-events A F I*  
**assumes**  $B \in F ' I$   
**shows** *indep-event A B*  
 $\langle$ *proof* $\rangle$

**lemma** *mutual-indep-ev-singleton-event2*:  
**assumes** *mutual-indep-events A F I*  
**assumes**  $i \in I$   
**shows** *indep-event A*  $(F i)$   
 $\langle$ *proof* $\rangle$

**lemma** *mutual-indep-iff*:  
**shows** *mutual-indep-events A F I*  $\longleftrightarrow$  *mutual-indep-set A*  $(F ' I)$   
 $\langle$ *proof* $\rangle$

**lemma** *mutual-indep-ev-cond*:  
**assumes**  $A \in \text{events}$  **and**  $F ' J \subseteq \text{events}$  **and** *finite J*  
**and** *mutual-indep-events A F I* **and**  $J \subseteq I$  **and**  $J \neq \{\}$  **and**  $\text{prob} (\bigcap (F ' J)) \neq 0$   
**shows**  $\mathcal{P}(A | (\bigcap (F ' J))) = \text{prob } A$   
 $\langle$ *proof* $\rangle$

**lemma** *mutual-indep-ev-cond-full*:  
**assumes**  $A \in \text{events}$  **and**  $F ' I \subseteq \text{events}$  **and** *finite I*  
**and** *mutual-indep-events A F I* **and**  $I \neq \{\}$  **and**  $\text{prob} (\bigcap (F ' I)) \neq 0$   
**shows**  $\mathcal{P}(A | (\bigcap (F ' I))) = \text{prob } A$   
 $\langle$ *proof* $\rangle$

**lemma** *mutual-indep-ev-cond-single*:  
**assumes**  $A \in \text{events}$  **and**  $B \in \text{events}$   
**and** *mutual-indep-events A F I* **and**  $B \in F ' I$  **and**  $\text{prob } B \neq 0$   
**shows**  $\mathcal{P}(A | B) = \text{prob } A$   
 $\langle$ *proof* $\rangle$

**lemma** *mutual-indep-ev-empty*:  $A \in \text{events} \implies$  *mutual-indep-events A F*  $\{\}$   
 $\langle$ *proof* $\rangle$

**lemma** *not-mutual-indep-ev-itself*:

**assumes**  $\text{prob } A > 0$  **and**  $\text{prob } A < 1$  **and**  $A = F\ i$

**shows**  $\neg \text{mutual-indep-events } A\ F\ \{i\}$

*<proof>*

**lemma** *is-mutual-indep-ev-itself*:

**assumes**  $A \in \text{events}$  **and**  $A = F\ i$

**assumes**  $\text{prob } A = 0 \vee \text{prob } A = 1$

**shows**  $\text{mutual-indep-events } A\ F\ \{i\}$

*<proof>*

**lemma** *mutual-indep-ev-singleton*:

**assumes**  $\text{indep-event } A\ (F\ i)$

**shows**  $\text{mutual-indep-events } A\ F\ \{i\}$

*<proof>*

**lemma** *mutual-indep-ev-one-compl*:

**assumes**  $\text{mutual-indep-events } A\ F\ I$

**assumes**  $\text{finite } I$

**assumes**  $i \in I$

**assumes**  $\text{space } M - F\ i = F\ j$

**shows**  $\text{mutual-indep-events } A\ F\ (\{j\} \cup I)$

*<proof>*

**lemma** *mutual-indep-events-update-compl*:

**assumes**  $\text{mutual-indep-events } X\ F\ S$

**assumes**  $S = A \cup B$

**assumes**  $A \cap B = \{\}$

**assumes**  $\text{finite } S$

**assumes**  $\text{bij-betw } G\ A\ A'$

**assumes**  $\bigwedge i. i \in A \implies F\ (G\ i) = \text{space } M - F\ i$

**shows**  $\text{mutual-indep-events } X\ F\ (A' \cup B)$

*<proof>*

**lemma** *mutual-indep-ev-events-compl*:

**assumes**  $\text{finite } S$

**assumes**  $\text{mutual-indep-events } A\ F\ S$

**assumes**  $\text{bij-betw } G\ S\ S'$

**assumes**  $\bigwedge i. i \in S \implies F\ (G\ i) = \text{space } M - F\ i$

**shows**  $\text{mutual-indep-events } A\ F\ S'$

*<proof>*

Important lemma on relation between independence and mutual independence of a set

**lemma** *mutual-indep-ev-set-all*:

**assumes**  $F\ 'I \subseteq \text{events}$

**assumes**  $\bigwedge i. i \in I \implies (\text{mutual-indep-events } (F\ i)\ F\ (I - \{i\}))$

**shows**  $\text{indep-events } F\ I$

*<proof>*

**end**  
**end**

## 6 The Basic Probabilistic Method Framework

This theory includes all aspects of step (3) and (4) of the basic method framework, which are purely probabilistic

**theory** *Basic-Method* **imports** *Indep-Events*  
**begin**

### 6.1 More Set and Multiset lemmas

**lemma** *card-size-set-mset*:  $\text{card } (\text{set-mset } A) \leq \text{size } A$

*<proof>*

**lemma** *Union-exists*:  $\{a \in A . \exists b \in B . P a b\} = (\bigcup b \in B . \{a \in A . P a b\})$

*<proof>*

**lemma** *Inter-forall*:  $B \neq \{\} \implies \{a \in A . \forall b \in B . P a b\} = (\bigcap b \in B . \{a \in A . P a b\})$

*<proof>*

**lemma** *function-map-multi-filter-size*:

**assumes** *image-mset*  $F$  (*mset-set*  $A$ ) =  $B$  **and** *finite*  $A$

**shows**  $\text{card } \{a \in A . P (F a)\} = \text{size } \{\# b \in \# B . P b \#\}$

*<proof>*

**lemma** *bij-mset-obtain-set-elem*:

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

**assumes**  $b \in \# B$

**obtains**  $a$  **where**  $a \in A$  **and**  $F a = b$

*<proof>*

**lemma** *bij-mset-obtain-mset-elem*:

**assumes** *finite*  $A$

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

**assumes**  $a \in A$

**obtains**  $b$  **where**  $b \in \# B$  **and**  $F a = b$

*<proof>*

**lemma** *prod-fn-le1*:

**fixes**  $f :: 'c \Rightarrow ('d :: \{\text{comm-monoid-mult, linordered-semidom}\})$

**assumes** *finite*  $A$

**assumes**  $A \neq \{\}$

**assumes**  $\bigwedge y. y \in A \implies f y \geq 0 \wedge f y < 1$

**shows**  $(\prod x \in A. f x) < 1$

*<proof>*

**context** *prob-space*  
**begin**

## 6.2 Existence Lemmas

**lemma** *prob-lt-one-obtain:*

**assumes**  $\{e \in \text{space } M . Q\ e\} \in \text{events}$   
**assumes**  $\text{prob } \{e \in \text{space } M . Q\ e\} < 1$   
**obtains**  $e$  **where**  $e \in \text{space } M$  **and**  $\neg Q\ e$

*<proof>*

**lemma** *prob-gt-zero-obtain:*

**assumes**  $\{e \in \text{space } M . Q\ e\} \in \text{events}$   
**assumes**  $\text{prob } \{e \in \text{space } M . Q\ e\} > 0$   
**obtains**  $e$  **where**  $e \in \text{space } M$  **and**  $Q\ e$

*<proof>*

**lemma** *inter-gt0-event:*

**assumes**  $F\ 'I \subseteq \text{events}$   
**assumes**  $\text{prob } (\bigcap i \in I . (\text{space } M - (F\ i))) > 0$   
**shows**  $(\bigcap i \in I . (\text{space } M - (F\ i))) \in \text{events}$  **and**  $(\bigcap i \in I . (\text{space } M - (F\ i))) \neq \{\}$

*<proof>*

**lemma** *obtain-intersection:*

**assumes**  $F\ 'I \subseteq \text{events}$   
**assumes**  $\text{prob } (\bigcap i \in I . (\text{space } M - (F\ i))) > 0$   
**obtains**  $e$  **where**  $e \in \text{space } M$  **and**  $\bigwedge i . i \in I \implies e \notin F\ i$

*<proof>*

**lemma** *obtain-intersection-prop:*

**assumes**  $F\ 'I \subseteq \text{events}$   
**assumes**  $\bigwedge i . i \in I \implies F\ i = \{e \in \text{space } M . P\ e\ i\}$   
**assumes**  $\text{prob } (\bigcap i \in I . (\text{space } M - (F\ i))) > 0$   
**obtains**  $e$  **where**  $e \in \text{space } M$  **and**  $\bigwedge i . i \in I \implies \neg P\ e\ i$

*<proof>*

**lemma** *not-in-big-union:*

**assumes**  $\bigwedge i . i \in A \implies e \notin i$   
**shows**  $e \notin (\bigcup A)$

*<proof>*

**lemma** *not-in-big-union-fn:*

**assumes**  $\bigwedge i . i \in A \implies e \notin F\ i$   
**shows**  $e \notin (\bigcup i \in A . F\ i)$

*<proof>*

**lemma** *obtain-intersection-union:*

**assumes**  $F' I \subseteq \text{events}$

**assumes**  $\text{prob} (\bigcap i \in I . (\text{space } M - (F' i))) > 0$

**obtains**  $e$  **where**  $e \in \text{space } M$  **and**  $e \notin (\bigcup i \in I . F' i)$

*<proof>*

### 6.3 Basic Bounds

Lemmas on the Complete Independence and Union bound

**lemma** *complete-indep-bound1:*

**assumes** *finite*  $A$

**assumes**  $A \neq \{\}$

**assumes**  $A \subseteq \text{events}$

**assumes** *indep-events-set*  $A$

**assumes**  $\bigwedge a . a \in A \implies \text{prob } a < 1$

**shows**  $\text{prob} (\text{space } M - (\bigcap A)) > 0$

*<proof>*

**lemma** *complete-indep-bound1-index:*

**assumes** *finite*  $A$

**assumes**  $A \neq \{\}$

**assumes**  $F' A \subseteq \text{events}$

**assumes** *indep-events*  $F' A$

**assumes**  $\bigwedge a . a \in A \implies \text{prob} (F' a) < 1$

**shows**  $\text{prob} (\text{space } M - (\bigcap (F' A))) > 0$

*<proof>*

**lemma** *complete-indep-bound2:*

**assumes** *finite*  $A$

**assumes**  $A \subseteq \text{events}$

**assumes** *indep-events-set*  $A$

**assumes**  $\bigwedge a . a \in A \implies \text{prob } a < 1$

**shows**  $\text{prob} (\text{space } M - (\bigcup A)) > 0$

*<proof>*

**lemma** *complete-indep-bound2-index:*

**assumes** *finite*  $A$

**assumes**  $F' A \subseteq \text{events}$

**assumes** *indep-events*  $F' A$

**assumes**  $\bigwedge a . a \in A \implies \text{prob} (F' a) < 1$

**shows**  $\text{prob} (\text{space } M - (\bigcup (F' A))) > 0$

*<proof>*

**lemma** *complete-indep-bound3:*

**assumes** *finite*  $A$

**assumes**  $A \neq \{\}$

**assumes**  $F' A \subseteq \text{events}$

**assumes** *indep-events*  $F' A$

**assumes**  $\bigwedge a . a \in A \implies \text{prob} (F' a) < 1$

**shows**  $\text{prob} (\bigcap a \in A. \text{space } M - F a) > 0$   
*<proof>*

Combining complete independence with existence step

**lemma** *complete-indep-bound-obtain:*  
**assumes** *finite A*  
**assumes**  $A \subseteq \text{events}$   
**assumes** *indep-events-set A*  
**assumes**  $\bigwedge a . a \in A \implies \text{prob } a < 1$   
**obtains**  $e$  **where**  $e \in \text{space } M$  **and**  $e \notin \bigcup A$   
*<proof>*

**lemma** *Union-bound-events:*  
**assumes** *finite A*  
**assumes**  $A \subseteq \text{events}$   
**shows**  $\text{prob} (\bigcup A) \leq (\sum a \in A. \text{prob } a)$   
*<proof>*

**lemma** *Union-bound-events-fun:*  
**assumes** *finite A*  
**assumes**  $f ' A \subseteq \text{events}$   
**shows**  $\text{prob} (\bigcup (f ' A)) \leq (\sum a \in A. \text{prob } (f a))$   
*<proof>*

**lemma** *Union-bound-avoid:*  
**assumes** *finite A*  
**assumes**  $(\sum a \in A. \text{prob } a) < 1$   
**assumes**  $A \subseteq \text{events}$   
**shows**  $\text{prob} (\text{space } M - \bigcup A) > 0$   
*<proof>*

**lemma** *Union-bound-avoid-fun:*  
**assumes** *finite A*  
**assumes**  $(\sum a \in A. \text{prob } (f a)) < 1$   
**assumes**  $f ' A \subseteq \text{events}$   
**shows**  $\text{prob} (\text{space } M - \bigcup (f ' A)) > 0$   
*<proof>*

Combining union bound with existence step

**lemma** *Union-bound-obtain:*  
**assumes** *finite A*  
**assumes**  $(\sum a \in A. \text{prob } a) < 1$   
**assumes**  $A \subseteq \text{events}$   
**obtains**  $e$  **where**  $e \in \text{space } M$  **and**  $e \notin \bigcup A$   
*<proof>*

**lemma** *Union-bound-obtain-fun:*  
**assumes** *finite A*



**assumes**  $(\sum a \in A. \text{prob } (f a)) < 1$   
**assumes**  $f' A \subseteq \text{events}$   
**obtains**  $e$  **where**  $e \in \text{space } M$  **and**  $e \notin \bigcup (f' A)$   
 $\langle \text{proof} \rangle$

**lemma** *Union-bound-obtain-compl:*  
**assumes** *finite*  $A$   
**assumes**  $(\sum a \in A. \text{prob } a) < 1$   
**assumes**  $A \subseteq \text{events}$   
**obtains**  $e$  **where**  $e \in (\text{space } M - \bigcup A)$   
 $\langle \text{proof} \rangle$

**lemma** *Union-bound-obtain-compl-fun:*  
**assumes** *finite*  $A$   
**assumes**  $(\sum a \in A. \text{prob } (f a)) < 1$   
**assumes**  $f' A \subseteq \text{events}$   
**obtains**  $e$  **where**  $e \in (\text{space } M - \bigcup (f' A))$   
 $\langle \text{proof} \rangle$

**end**

**end**

## 7 Lovasz Local Lemma

**theory** *Lovasz-Local-Lemma*  
**imports**  
*Basic-Method*  
*HOL-Real-Asymp.Real-Asymp*  
*Indep-Events*  
*Digraph-Extensions*  
**begin**

**lemma** *ln-add-one-self-less-self:*  
**fixes**  $x :: \text{real}$   
**assumes**  $x > 0$   
**shows**  $\ln (1 + x) < x$   
 $\langle \text{proof} \rangle$

**lemma** *exp-1-bounds:*  
**assumes**  $x > (0 :: \text{real})$   
**shows**  $\exp 1 > (1 + 1 / x) \text{ powr } x$  **and**  $\exp 1 < (1 + 1 / x) \text{ powr } (x+1)$   
 $\langle \text{proof} \rangle$

### 7.1 Random Lemmas on Product Operator

**lemma** *prod-constant-ge:*  
**fixes**  $y :: 'b :: \{\text{comm-monoid-mult, linordered-semidom}\}$

**assumes**  $\text{card } A \leq k$   
**assumes**  $y \geq 0$  **and**  $y < 1$   
**shows**  $(\prod_{x \in A} y) \geq y^{\wedge k}$   
 <proof>

**lemma** (in *linordered-idom*) *prod-mono3*:  
**assumes** *finite*  $J$   $I \subseteq J$   $\bigwedge i. i \in J \implies 0 \leq f i$  ( $\bigwedge i. i \in J \implies f i \leq 1$ )  
**shows**  $\text{prod } f J \leq \text{prod } f I$   
 <proof>

**lemma** *bij-on-ss-image*:  
**assumes**  $A \subseteq B$   
**assumes** *bij-betw*  $g$   $B$   $B'$   
**shows**  $g \text{ ` } A \subseteq B'$   
 <proof>

**lemma** *bij-on-ss-proper-image*:  
**assumes**  $A \subset B$   
**assumes** *bij-betw*  $g$   $B$   $B'$   
**shows**  $g \text{ ` } A \subset B'$   
 <proof>

## 7.2 Dependency Graph Concept

Uses directed graphs. The *pair\_digraph* locale was sufficient as multi-edges are irrelevant

**locale** *dependency-digraph* = *pair-digraph*  $G :: \text{nat pair-pre-digraph} + \text{prob-space}$   
 $M :: \text{'a measure}$   
**for**  $G$   $M$  + **fixes**  $F :: \text{nat} \implies \text{'a set}$   
**assumes** *vss*:  $F \text{ ` } (\text{pverts } G) \subseteq \text{events}$   
**assumes** *mis*:  $\bigwedge i. i \in (\text{pverts } G) \implies \text{mutual-indep-events } (F i) F ((\text{pverts } G) - (\{i\} \cup \text{neighborhood } i))$   
**begin**

**lemma** *dep-graph-indiv-nh-indep*:  
**assumes**  $A \in \text{pverts } G$   $B \in \text{pverts } G$   
**assumes**  $B \notin \text{neighborhood } A$   
**assumes**  $A \neq B$   
**assumes**  $\text{prob } (F B) \neq 0$   
**shows**  $\mathcal{P}((F A) \mid (F B)) = \text{prob } (F A)$   
 <proof>

**lemma** *mis-subset*:  
**assumes**  $i \in \text{pverts } G$   
**assumes**  $A \subseteq \text{pverts } G$   
**shows**  $\text{mutual-indep-events } (F i) F (A - (\{i\} \cup \text{neighborhood } i))$   
 <proof>

**lemma** *dep-graph-indep-events*:

**assumes**  $A \subseteq pverts\ G$   
**assumes**  $\bigwedge Ai. Ai \in A \implies out-degree\ G\ Ai = 0$   
**shows**  $indep-events\ F\ A$   
 <proof>

**end**

### 7.3 Lovasz Local General Lemma

**context**  $prob-space$

**begin**

**lemma** *compl-sets-index:*

**assumes**  $F \text{ ' } A \subseteq events$   
**shows**  $(\lambda i. space\ M - F\ i) \text{ ' } A \subseteq events$   
 <proof>

**lemma** *lovasz-inductive-base:*

**assumes**  $dependency-digraph\ G\ M\ F$   
**assumes**  $\bigwedge Ai. Ai \in A \implies g\ Ai \geq 0 \wedge g\ Ai < 1$   
**assumes**  $\bigwedge Ai. Ai \in A \implies (prob\ (F\ Ai) \leq (g\ Ai) * (\prod_{Aj \in pre-digraph.neighborhood\ G\ Ai} (1 - (g\ Aj))))$   
**assumes**  $Ai \in A$   
**assumes**  $pverts\ G = A$   
**shows**  $prob\ (F\ Ai) \leq g\ Ai$   
 <proof>

**lemma** *lovasz-inductive-base-set:*

**assumes**  $N \subseteq A$   
**assumes**  $\bigwedge Ai. Ai \in A \implies g\ Ai \geq 0 \wedge g\ Ai < 1$   
**assumes**  $\bigwedge Ai. Ai \in A \implies (prob\ (F\ Ai) \leq (g\ Ai) * (\prod_{Aj \in N} (1 - (g\ Aj))))$   
**assumes**  $Ai \in A$   
**shows**  $prob\ (F\ Ai) \leq g\ Ai$   
 <proof>

**lemma** *split-prob-lt-helper:*

**assumes**  $dep-graph: dependency-digraph\ G\ M\ F$   
**assumes**  $dep-graph-verts: pverts\ G = A$   
**assumes**  $fbounds: \bigwedge i. i \in A \implies f\ i \geq 0 \wedge f\ i < 1$   
**assumes**  $prob-Ai: \bigwedge Ai. Ai \in A \implies prob\ (F\ Ai) \leq$   
 $(f\ Ai) * (\prod_{Aj \in pre-digraph.neighborhood\ G\ Ai} (1 - (f\ Aj)))$   
**assumes**  $aiin: Ai \in A$   
**assumes**  $N \subseteq pre-digraph.neighborhood\ G\ Ai$   
**assumes**  $\exists P1\ P2. \mathcal{P}(F\ Ai \mid \bigcap_{Aj \in S} space\ M - F\ Aj) = P1/P2 \wedge$   
 $P1 \leq prob\ (F\ Ai) \wedge P2 \geq (\prod_{Aj \in N} (1 - (f\ Aj)))$   
**shows**  $\mathcal{P}(F\ Ai \mid \bigcap_{Aj \in S} space\ M - F\ Aj) \leq f\ Ai$   
 <proof>

**lemma** *lovasz-inequality:*

**assumes** *finS*: *finite S*  
**assumes** *sevents*:  $F \text{ ' } S \subseteq \text{events}$   
**assumes** *S-subset*:  $S \subseteq A - \{Ai\}$   
**assumes** *prob2*:  $\text{prob} (\bigcap Aj \in S . (\text{space } M - (F Aj))) > 0$   
**assumes** *irange*:  $i \in \{0..<\text{card } S1\}$   
**assumes** *bb*: *bij-betw g {0..<card S1} S1*  
**assumes** *s1-def*:  $S1 = (S \cap N)$   
**assumes** *s2-def*:  $S2 = S - S1$   
**assumes** *ne-cond*:  $i > 0 \vee S2 \neq \{\}$   
**assumes** *hyps*:  $\bigwedge B. B \subseteq S \implies g i \in A \implies B \subseteq A - \{g i\} \implies B \neq \{\} \implies$   
 $0 < \text{prob} (\bigcap Aj \in B. \text{space } M - F Aj) \implies \mathcal{P}(F (g i) \mid \bigcap Aj \in B. \text{space } M - F$   
 $Aj) \leq f (g i)$   
**shows**  $\mathcal{P}((\text{space } M - F (g i)) \mid (\bigcap ((\lambda i. \text{space } M - F i) \text{ ' } g \text{ ' } \{0..<i\} \cup ((\lambda i. \text{space } M - F i) \text{ ' } S2))))$   
 $\geq (1 - f (g i))$   
*<proof>*

The main helper lemma

**lemma** *lovasz-inductive*:

**assumes** *finA*: *finite A*  
**assumes** *Aevents*:  $F \text{ ' } A \subseteq \text{events}$   
**assumes** *fbounds*:  $\bigwedge i . i \in A \implies f i \geq 0 \wedge f i < 1$   
**assumes** *dep-graph*: *dependency-digraph G M F*  
**assumes** *dep-graph-verts*: *pverts G = A*  
**assumes** *prob-Ai*:  $\bigwedge Ai . Ai \in A \implies \text{prob} (F Ai) \leq$   
 $(f Ai) * (\prod Aj \in \text{pre-digraph.neighborhood } G Ai . (1 - (f Aj)))$   
**assumes** *Ai-in*:  $Ai \in A$   
**assumes** *S-subset*:  $S \subseteq A - \{Ai\}$   
**assumes** *S-nempty*:  $S \neq \{\}$   
**assumes** *prob2*:  $\text{prob} (\bigcap Aj \in S . (\text{space } M - (F Aj))) > 0$   
**shows**  $\mathcal{P}((F Ai) \mid (\bigcap Aj \in S . (\text{space } M - (F Aj)))) \leq f Ai$   
*<proof>*

The main lemma

**theorem** *lovasz-local-general*:

**assumes**  $A \neq \{\}$   
**assumes**  $F \text{ ' } A \subseteq \text{events}$   
**assumes** *finite A*  
**assumes**  $\bigwedge Ai . Ai \in A \implies f Ai \geq 0 \wedge f Ai < 1$   
**assumes** *dependency-digraph G M F*  
**assumes**  $\bigwedge Ai . Ai \in A \implies (\text{prob} (F Ai) \leq (f Ai) * (\prod Aj \in \text{pre-digraph.neighborhood } G Ai . (1 - (f Aj))))$   
**assumes** *pverts G = A*  
**shows**  $\text{prob} (\bigcap Ai \in A . (\text{space } M - (F Ai))) \geq (\prod Ai \in A . (1 - f Ai)) (\prod$   
 $Ai \in A . (1 - f Ai)) > 0$   
*<proof>*

## 7.4 Lovasz Corollaries and Variations

**corollary** *lovasz-local-general-positive*:

**assumes**  $A \neq \{\}$   
**assumes**  $F \text{ ' } A \subseteq \text{events}$   
**assumes**  $\text{finite } A$   
**assumes**  $\bigwedge Ai . Ai \in A \implies f Ai \geq 0 \wedge f Ai < 1$   
**assumes**  $\text{dependency-digraph } G M F$   
**assumes**  $\bigwedge Ai . Ai \in A \implies (\text{prob } (F Ai) \leq$   
 $(f Ai) * (\prod Aj \in \text{pre-digraph.neighborhood } G Ai . (1 - (f Aj))))$   
**assumes**  $\text{pverts } G = A$   
**shows**  $\text{prob } (\bigcap Ai \in A . (\text{space } M - (F Ai))) > 0$   
 $\langle \text{proof} \rangle$

**theorem** *lovasz-local-symmetric-dep-graph:*

**fixes**  $e :: \text{real}$   
**fixes**  $d :: \text{nat}$   
**assumes**  $A \neq \{\}$   
**assumes**  $F \text{ ' } A \subseteq \text{events}$   
**assumes**  $\text{finite } A$   
**assumes**  $\text{dependency-digraph } G M F$   
**assumes**  $\bigwedge Ai . Ai \in A \implies \text{out-degree } G Ai \leq d$   
**assumes**  $\bigwedge Ai . Ai \in A \implies \text{prob } (F Ai) \leq p$   
**assumes**  $\exp(1) * p * (d + 1) \leq 1$   
**assumes**  $\text{pverts } G = A$   
**shows**  $\text{prob } (\bigcap Ai \in A . (\text{space } M - (F Ai))) > 0$   
 $\langle \text{proof} \rangle$

**corollary** *lovasz-local-symmetric4gt:*

**fixes**  $e :: \text{real}$   
**fixes**  $d :: \text{nat}$   
**assumes**  $A \neq \{\}$   
**assumes**  $F \text{ ' } A \subseteq \text{events}$   
**assumes**  $\text{finite } A$   
**assumes**  $\text{dependency-digraph } G M F$   
**assumes**  $\bigwedge Ai . Ai \in A \implies \text{out-degree } G Ai \leq d$   
**assumes**  $\bigwedge Ai . Ai \in A \implies \text{prob } (F Ai) \leq p$   
**assumes**  $4 * p * d \leq 1$   
**assumes**  $d \geq 3$   
**assumes**  $\text{pverts } G = A$   
**shows**  $\text{prob } (\bigcap Ai \in A . (\text{space } M - F Ai)) > 0$   
 $\langle \text{proof} \rangle$

**lemma** *lovasz-local-symmetric4:*

**fixes**  $e :: \text{real}$   
**fixes**  $d :: \text{nat}$   
**assumes**  $A \neq \{\}$   
**assumes**  $F \text{ ' } A \subseteq \text{events}$   
**assumes**  $\text{finite } A$   
**assumes**  $\text{dependency-digraph } G M F$   
**assumes**  $\bigwedge Ai . Ai \in A \implies \text{out-degree } G Ai \leq d$

**assumes**  $\bigwedge Ai. Ai \in A \implies \text{prob } (F Ai) \leq p$   
**assumes**  $4 * p * d \leq 1$   
**assumes**  $d \geq 1$   
**assumes**  $\text{pverts } G = A$   
**shows**  $\text{prob } (\bigcap Ai \in A . (\text{space } M - F Ai)) > 0$   
 <proof>

Converting between dependency graph and indexed set representation of mutual independence

**lemma** (in *pair-digraph*) *g-Ai-simplification*:

**assumes**  $Ai \in A$   
**assumes**  $g Ai \subseteq A - \{Ai\}$   
**assumes**  $\text{pverts } G = A$   
**assumes**  $\text{parcs } G = \{e \in A \times A . \text{snd } e \in (A - (\{fst e\} \cup (g (fst e))))\}$   
**shows**  $g Ai = A - (\{Ai\} \cup \text{neighborhood } Ai)$   
 <proof>

**lemma** *define-dep-graph-set*:

**assumes**  $A \neq \{\}$   
**assumes**  $F ' A \subseteq \text{events}$   
**assumes** *finite*  $A$   
**assumes**  $\bigwedge Ai. Ai \in A \implies g Ai \subseteq A - \{Ai\} \wedge \text{mutual-indep-events } (F Ai) F (g Ai)$   
**shows** *dependency-digraph* ( $\text{pverts} = A, \text{parcs} = \{e \in A \times A . \text{snd } e \in (A - (\{fst e\} \cup (g (fst e))))\}$ )  $\Downarrow$   $M F$   
 (is *dependency-digraph* ? $G M F$ )  
 <proof>

**lemma** *define-dep-graph-deg-bound*:

**assumes**  $A \neq \{\}$   
**assumes**  $F ' A \subseteq \text{events}$   
**assumes** *finite*  $A$   
**assumes**  $\bigwedge Ai. Ai \in A \implies g Ai \subseteq A - \{Ai\} \wedge \text{card } (g Ai) \geq \text{card } A - d - 1$   
 $\wedge$   
 $\text{mutual-indep-events } (F Ai) F (g Ai)$   
**shows**  $\bigwedge Ai. Ai \in A \implies$   
 $\text{out-degree } (\text{pverts} = A, \text{parcs} = \{e \in A \times A . \text{snd } e \in (A - (\{fst e\} \cup (g (fst e))))\}) \Downarrow Ai \leq d$   
 (is  $\bigwedge Ai. Ai \in A \implies \text{out-degree } (\text{with-proj } ?G) Ai \leq d$ )  
 <proof>

**lemma** *obtain-dependency-graph*:

**assumes**  $A \neq \{\}$   
**assumes**  $F ' A \subseteq \text{events}$   
**assumes** *finite*  $A$   
**assumes**  $\bigwedge Ai. Ai \in A \implies$   
 $(\exists S . S \subseteq A - \{Ai\} \wedge \text{card } S \geq \text{card } A - d - 1 \wedge \text{mutual-indep-events } (F Ai) F S)$   
**obtains**  $G$  where *dependency-digraph*  $G M F \text{pverts } G = A \bigwedge Ai. Ai \in A \implies$

*out-degree*  $G \text{ Ai} \leq d$   
 ⟨*proof*⟩

This is the variation of the symmetric version most commonly in use

**theorem** *lovasz-local-symmetric*:

**fixes**  $d :: \text{nat}$   
**assumes**  $A \neq \{\}$   
**assumes**  $F \text{ ' } A \subseteq \text{events}$   
**assumes** *finite*  $A$   
**assumes**  $\bigwedge \text{Ai. Ai} \in A \implies (\exists S . S \subseteq A - \{\text{Ai}\} \wedge \text{card } S \geq \text{card } A - d - 1$   
 $\wedge \text{mutual-indep-events } (F \text{ Ai}) F S)$   
**assumes**  $\bigwedge \text{Ai. Ai} \in A \implies \text{prob } (F \text{ Ai}) \leq p$   
**assumes**  $\exp(1) * p * (d + 1) \leq 1$   
**shows**  $\text{prob } (\bigcap \text{Ai} \in A . (\text{space } M - (F \text{ Ai}))) > 0$   
 ⟨*proof*⟩

**lemma** *lovasz-local-symmetric4-set*:

**fixes**  $d :: \text{nat}$   
**assumes**  $A \neq \{\}$   
**assumes**  $F \text{ ' } A \subseteq \text{events}$   
**assumes** *finite*  $A$   
**assumes**  $\bigwedge \text{Ai. Ai} \in A \implies (\exists S . S \subseteq A - \{\text{Ai}\} \wedge \text{card } S \geq \text{card } A - d - 1$   
 $\wedge \text{mutual-indep-events } (F \text{ Ai}) F S)$   
**assumes**  $\bigwedge \text{Ai. Ai} \in A \implies \text{prob } (F \text{ Ai}) \leq p$   
**assumes**  $4 * p * d \leq 1$   
**assumes**  $d \geq 1$   
**shows**  $\text{prob } (\bigcap \text{Ai} \in A . (\text{space } M - F \text{ Ai})) > 0$   
 ⟨*proof*⟩

**end**

**end**

**theory** *Lovasz-Local-Root*

**imports**

*PiE-Rel-Extras*

*Digraph-Extensions*

*Prob-Events-Extras*

*Cond-Prob-Extensions*

*Indep-Events*

*Basic-Method*

*Lovasz-Local-Lemma*

**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] L. Noschinski. A Graph Library for Isabelle. *Mathematics in Computer Science*, 9(1):23–39, Mar. 2015.
- [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/).